

ORGANIZANDO MIS PRÁCTICAS

ADRIANA VENTURA CANDELA

TRABAJO DE FIN DE GRADO EN INGENIERÍA
INFORMÁTICA

FACULTAD DE INFORMÁTICA

UNIVERSIDAD COMPLUTENSE DE MADRID



CURSO 2017-2018

Director: RUBÉN FUENTES FERNÁNDEZ

Agradecimientos

Este trabajo representa el final de un largo camino y el comienzo de uno nuevo. Quiero dar las gracias a mis padres, abuelos y familia que han hecho posible con su esfuerzo y paciencia que hoy pueda finalizar esta etapa de mi vida.

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS	5
ÍNDICE DE TABLAS	6
RESUMEN.....	7
DESCRIPTORES.....	7
ABSTRACT	8
1. INTRODUCCIÓN	9
1.1 Motivación	9
1.2 Objetivos y plan de trabajo	9
INTRODUCTION.....	11
Motivation	11
Goals and work plan	11
2. ESTADO DEL ARTE.....	13
2.1 Campus Virtual UCM	13
2.2 Calendario	14
2.3 UCMovil	15
2.4 Requisitos de la aplicación y comparativa	15
3. ARQUITECTURA DE LA APLICACIÓN.....	18
3.1 Arquitectura cliente-servidor	18
3.2 Módulos del sistema.....	18
3.2.1 Diagrama de clases.....	19
3.2.2 Diagrama de secuencia	20
3.3 Consideraciones de implementación.....	28
3.4 Lenguajes y herramientas de desarrollo.....	29
4. MODELO DE DATOS	31
5. DISEÑO INTERFAZ DE USUARIO.....	34
6. FUNCIONALIDAD DE LA APLICACIÓN.....	38
6.1 Manual del administrador	38
6.1.1 Login.....	38
6.1.2 Página principal	38
6.1.3 Editar contraseña.....	40
6.1.4 Mostrar lista de datos.....	40
6.1.5 Crear un nuevo dato.....	41
6.1.6 Ver información sobre un dato	42
6.1.7 Editar información de un dato	43
6.1.8 Borrar un dato	44
6.2 Manual del profesorado.....	46
6.2.1 Login.....	46
6.2.2 Página principal	46
6.2.3 Crea tarea	47
6.2.4 Editar contraseña.....	47
6.2.5 Lista de tareas terminadas.....	47
6.2.6 Evaluar tarea	48
6.2.7 Ver lista de asignaturas y ver lista de alumnos.....	49
6.2.9 Información personal	50
6.3 Manual del estudiante	51
6.3.1 Login.....	51

6.3.2	Página principal	51
6.3.3	Ver información de una tarea pendiente.....	53
6.3.4	Ver lista de tareas terminadas	53
6.3.5	Ver información de una tarea terminada.....	54
6.3.6	Ver lista de asignaturas y ver lista de profesores.....	55
6.3.7	Ver información de una asignatura y ver información de un profesor	56
6.3.8	Editar contraseña.....	57
6.3.9	Datos personales	57
7.	CONCLUSIONES Y TRABAJO FUTURO	58
7.1	Conclusiones	58
7.2	Trabajo futuro.....	59
	CONCLUSIONS AND FUTURE WORK	60
	CONCLUSIONS	60
	FUTURE WORK	61
8.	Bibliografía	62

ÍNDICE DE FIGURAS

Figura 1: Visualización de una tarea en Campus Virtual UCM.....	13
Figura 2: Visualización de una tarea en Calendario.....	14
Figura 3: Diagrama de clases	19
Figura 4: Diagrama secuencia – Login	21
Figura 5: Diagrama secuencia - Editar contraseña.....	23
Figura 6: Diagrama secuencia - Ver toda la información	24
Figura 7: Diagrama secuencia - Ver información de un dato concreto.....	25
Figura 8: Diagrama secuencia - Editar dato	26
Figura 9: Diagrama secuencia - Eliminar dato.....	27
Figura 10: Diagrama secuencia - Crear dato.....	28
Figura 11: Diagrama entidad-relación.....	33
Figura 12: Boceto web - Lista de elementos	35
Figura 13: Boceto web - Ver información de un elemento	36
Figura 14: Boceto APP - Lista de elementos y ver elemento.....	36
Figura 15: Log in - web.....	38
Figura 16: Página principal administrador web	39
Figura 17: Botón menú y botón salir.....	39
Figura 18: Página "Editar contraseña" - web	40
Figura 19: Botón home.....	40
Figura 20: Página "Lista de elementos" - web	41
Figura 21: Acciones sobre un elemento	41
Figura 22: Página "Crear elemento" - web.....	42
Figura 23: Error sobre tipo de datos - web.....	42
Figura 24: Error restrictivo - web.....	42
Figura 25: Página "Ver un elemento" - web	43
Figura 26: Página "Editar elemento" - web.....	43
Figura 27: Mensaje confirmación para borrar un elemento	44
Figura 28: Página "Borrar un elemento" - web	44
Figura 29: Página principal web del profesor - web	46
Figura 30: Página "Lista tareas terminadas" - web	48
Figura 31: Botón evaluar.....	48
Figura 32: Página "Evaluar tareas" - web	48
Figura 33: Log in - móvil	51
Figura 34: Página principal - móvil	52
Figura 35: Menú - móvil	52
Figura 36: Ver tarea pendiente - móvil	53
Figura 37: Tarea terminada - móvil.....	54
Figura 38: Lista asignaturas - móvil.....	55
Figura 39: Lista profesores - móvil.....	55
Figura 40: Ver asignatura - móvil	56

ÍNDICE DE TABLAS

Tabla 1: Comparativa entre aplicaciones existentes y requisitos de la aplicación.....	16
Tabla 2: Métodos comunes de clases	20

RESUMEN

El presente trabajo tiene su origen en la necesidad de herramientas de apoyo para la organización y visualización de las tareas de los alumnos a lo largo del curso. Se ha optado por una aplicación para dispositivos móviles destinada a los estudiantes. La aplicación permite la visualización de las tareas pendientes y terminadas durante el curso, así como la información de dichas tareas con sus respectivas calificaciones, la información de la asignatura a la que pertenece cada tarea y la información del profesor. También se desarrollan dos web para el mantenimiento de la aplicación por parte del administrador y el profesorado. El administrador se encarga de realizar una gestión de la base de datos. Los profesores se encargan de realizar una gestión de las entregas y notas de las tareas de cada una de sus asignaturas que tienen a lo largo del curso.

En la actualidad existen aplicaciones que gestionan parcialmente esta funcionalidad, pero presentan limitaciones. El objetivo de este trabajo es mejorar la relación de los estudiantes con la organización de las tareas que se presentan durante el curso. La organización es importante para que el alumno no se sienta saturado con el trabajo y pueda lograr cumplir con los plazos previstos y mejorar sus resultados.

Para desarrollar la aplicación, se utiliza un modelo cliente-servidor para mejorar sus capacidades funcionales y así poder cumplir con los requisitos establecidos.

DESCRIPTORES

1. Aplicación móvil
2. Estudiantes
3. Organización
4. Calendario
5. Tareas
6. Recordatorio
7. Cliente – Servidor
8. Apache Cordova

ABSTRACT

The present work has its origin in the need of support tools for the organization and visualization of the tasks of the students throughout the course. It has opted for an application for mobile devices for students. The application allows the visualization of the pending and finished tasks during the course, as well as the information of those tasks with their respective qualifications, the information of the subject to which each task belongs and the teacher's information. It can also be used for maintenance of the application by the administrator and the teachers. The administrator is responsible for managing the database. The teachers are responsible for managing the delivery and notes of the tasks each of their subjects they have throughout the course.

At present, there are applications that partially manage this functionality, but present limitations. The objective of this work is to improve the relationship of students with the organization of the tasks that are presented during the course. The organization is important so that the student does not feel saturated with the work and can achieve the expected deadlines and improve their results.

To develop the application, a client-server model is used to improve its functional capabilities and thus be able to meet the established requirements.

DESCRIPTORS

1. Mobile application
2. Students
3. Organization
4. Calendar
5. Tasks
6. Reminder
7. Client - Server
8. Apache Cordova

1. INTRODUCCIÓN

En esta sección se especifica la motivación para la realización del proyecto, los objetivos perseguidos a la hora de realizarlo a raíz de dicha motivación y el plan de trabajo llevado a cabo durante el proceso.

1.1 Motivación

La organización de las tareas del curso, puede resultar un ardua tarea para los alumnos. La media de asignaturas matriculas por año académico es de seis a doce por alumno. Cada asignatura puede poner distintos ejercicios, proyectos o exámenes a realizar durante el período académico. Muchas tareas no se llegan a presentar o la nota de estas puede disminuir considerablemente con respecto a la media. Esta situación puede provocar que los alumnos se sientan saturados y sometidos a estrés por la entrega de los trabajos. Dicho estrés se produce por la dificultad de la tarea en sí y por la falta de organización por parte del alumnado. El objetivo de este trabajo es ayudar al alumno con este último punto.

1.2 Objetivos y plan de trabajo

La funcionalidad del trabajo se divide en dos partes, introducción de contenidos y organización sobre ellos. Los contenidos relacionados con las tareas de los estudiantes se gestionan por el profesorado de la respectivas asignaturas. El contenido relacionado con la base de datos, como por ejemplo los alumnos de cada universidad, se gestiona por el administrador. El sistema se encarga de organizar los contenidos, y priorizar la realización de las tareas en función del tiempo restante de la fecha límite de entrega. El sistema se encarga de mostrar la organización a los alumnos.

Se desea realizar la aplicación con una arquitectura cliente-servidor. El servidor será capaz de atender las peticiones de los clientes, como por ejemplo, la creación de una nueva tarea. Los clientes serán las páginas web para el administrador y el profesorado, y la aplicación para dispositivos móviles. Se utiliza un desarrollo multiplataforma para la aplicación, que se adapta automáticamente a diferentes características del dispositivo del cliente.

Este proyecto se basa en la construcción de una aplicación para móviles, llamada OrganizUcm, siguiendo las pautas aprendidas durante el grado y las técnicas de desarrollo móvil. Cabe destacar que se desea realizar un diseño limpio y claro, para que los respectivos usuarios tengan una relación amigable con los contenidos de la aplicación correspondiente.

Para abordar los objetivos y requisitos previos, el trabajo se ha dividido en las siguientes tareas:

1. *Estudio del estado del arte* en cuanto a las aplicaciones existentes para los estudiantes. El objetivo es determinar las utilidades presentes en dichas aplicaciones e identificar cuales son las tendencias de diseño en cuanto interfaces de usuario. En esta etapa se procederá también a la selección de las tecnologías

necesarias para la creación de la aplicación móvil y la persistencia de los datos. Dicha información se detalla en la bibliografía existente, como se puede observar en el apartado 5. El tiempo estimado para la realización de esta tarea es de un mes, donde se incluye la instalación del material que se utilizará para realizar dicha tarea.

2. *Diseño y construcción de la base de datos.* Se incluye el estudio de los datos utilizados con los que trabajan las distintas aplicaciones, el diseño del diagrama entidad-relación y la inicialización de la base de datos en el gestor seleccionado. Se estima una duración de dos semanas para realizar este proceso.
3. *Generación de las consultas* a la base de datos que se utilizarán en la aplicación. Se tendrán en cuenta las consultas que pueden necesitar los distintos usuarios a la hora de utilizar las distintas aplicaciones. Se estima una semana para su realización.
4. *Diseño de un prototipo inicial* con herramientas de prototipado rápido para las distintas interfaces de usuario teniendo en cuenta los detalles de usabilidad y la claridad en las aplicaciones. La realización de dichos bocetos se prevé que tendrá un tiempo estimado de tres a cuatro semanas.
5. *Desarrollo de la funcionalidad de la aplicación.* Se desarrollará el servidor en relación con las consultas a la base de datos, orientado a las tareas que necesitarán las futuras aplicaciones, realizando las pruebas oportunas. Y se realizarán los distintos clientes que formarán el conjunto de la aplicación. Se estima una duración de dos meses para la realización de ambas.
6. *Fase de elaboración de la memoria.* Se procederá a escribir el documento que reunirá toda la información del proyecto realizado. Esta última fase se estima que se realizará en un mes y medio.

INTRODUCTION

This section specifies the motivation for the realization of the project, the objectives pursued when carrying out this project as a result of the motivation and the work plan carried out during the project process.

Motivation

The organization of the tasks of the course can be an arduous task for the students. The average number of subjects per academic year is six to twelve per student. Each subject can put different exercises, projects or exams to be done during the academic period. Many tasks are not presented or the note of these can decrease considerably with respect to the average. This situation can cause students to feel saturated and stressed by the delivery of work. This stress is caused by the difficulty of the task itself and by the lack of organization by the students. The objective of this work is to help the student with this last point.

Goals and work plan

The functionality of the work is divided into two parts, content introduction and organization over them. The contents related to the tasks of the students are managed by the faculty of the respective subjects. The content related to the database, such as the students of each university, is managed by the administrator. The system is responsible for organizing the contents, and prioritize the completion of tasks based on the time remaining from the delivery date. The system is responsible for showing the organization to the students.

You want to make the application with a client-server architecture. The server will be able to meet customer requests, such as the creation of a new task. The clients will be the web pages for the administrator and the teachers and the application for mobile devices. A cross-platform development is used for the application, which automatically adapts to different characteristics of the client's device.

This project is based on the construction of a mobile application, called OrganizUcm, following the guidelines learned during the degree and mobile development techniques. It should be noted that you want to make a clean and clear design, so that the respective users have a friendly relationship with the contents of the corresponding application.

To address the objectives and prerequisites, the work has been divided into the following tasks:

1. *Study of the state of the art* in terms of existing applications for students. The objective is to determine the utilities present in these applications and identify which are the design trends as user interfaces. At this stage we will also proceed with the selection of the necessary technologies for the creation of the mobile application and the persistence of the data. This information is detailed in the existing bibliography, as can be seen in section 5. The estimated time for the

completion of this task is one month, which includes the installation of the material that will be used to perform this task.

2. *Design and construction of the database.* It includes the study of the data used with which the different applications work, the design of the entity-relationship diagram and the initialization of the database in the selected manager. It is estimated a duration of two weeks to carry out this process.
3. *Generation of queries* to the database that will be used in the application. The queries that different users may need when using the different applications will be taken into account. It is estimated a week for its completion.
4. *Design of an initial prototype* with rapid prototyping tools for the different user interfaces taking into account the details of usability and clarity in the applications. The realization of said sketches is expected to take an estimated time of three to four weeks.
5. *Development of the functionality of the application.* The server will be developed in relation to queries to the database, oriented to the tasks that future applications will need, performing the appropriate tests. And the different clients that will form the whole of the application will be made. It is estimated a duration of two months for the realization of both.
6. *Phase of elaboration of memory.* We will proceed to write the document that will gather all the information of the project carried out. This last phase is estimated to take place in a month and a half.

2. ESTADO DEL ARTE

Antes de comenzar a implementar el proyecto, se ha realizado un estudio de las distintas aplicaciones que son utilizadas por los estudiantes de la universidad. En este apartado se va a hablar de dichas aplicaciones y las ventajas y desventajas que ofrece cada una de ellas. Finalmente, se van a establecer unos requisitos mínimos, obtenidos gracias a dicho estudio, que deberá tener la aplicación principal, es decir, la aplicación para dispositivos que utilizarán los estudiantes.

2.1 Campus Virtual UCM

Es un conjunto de espacios y herramientas en Internet que sirven de apoyo al aprendizaje de los alumnos. Se puede acceder mediante un navegador web, ya sea desde un ordenador o desde un dispositivo móvil. Ofrece las siguientes posibilidades [1] :

1. *Gestión de alumnos y grupos de trabajo.* Los profesores pueden conocer cuales son los alumnos matriculados en cada una de sus asignaturas. Así como gestionar los grupos de trabajo de los alumnos.
2. *Comunicación con el alumnado.* Se utiliza para abrir foros de debate entre los alumnos y el profesor, así como información de charlas y anuncios relacionados con la asignatura o con algún evento de interés en el ámbito de la carrera. También ofrece la posibilidad de gestionar el correo y la agenda de trabajo.
3. *Organización de contenidos.* Son documentos útiles para la asignatura, como el contenido del temario o la información de alguna tarea, como puede ser un manual para la realización.
4. *Envío, recepción y evaluación de tareas.* Se muestran en el apartado ‘Calificaciones’ y aparece la evaluación de tareas de un alumno en una asignatura determinada.

En la Figura 1, se puede observar como se establece la información de una tarea en el Campus Virtual UCM.

Entrega Final del proyecto	
Entrega Final del proyecto	
Estado de la entrega	
Grupo	G05
Número del intento	Este es el intento 1.
Estado de la entrega	No se ha enviado nada en esta tarea
Estado de la calificación	Sin calificar
Fecha de entrega	viernes, 15 de junio de 2018, 23:50
Tiempo restante	21 días 12 horas
Última modificación	-
Comentarios de la entrega	» Comentarios (0)
Agregar entrega	

Figura 1: Visualización de una tarea en Campus Virtual UCM

Como se puede observar, pese a ser un elemento muy completo para los estudiantes, no ofrece la posibilidad de establecer una organización de las tareas del alumno, lo que no solucionaría el problema planteado. Igualmente, aunque ofrece la posibilidad de ver las tareas que tiene el alumno, solo está disponible para observarlas de manera individual y accediendo a cada asignatura, no ofrece la posibilidad de verlas de manera global. Además, no está disponible para dispositivos móviles, por lo que la tarea puede implicar que los alumnos solo la utilicen desde sus móviles cuando se considera que es estrictamente necesario.

2.2 Calendario

Es un sistema de división del tiempo por días, semanas, meses y años. Ofrece la posibilidad de ver las tareas próximas de una asignatura determinada o la posibilidad de exportar el calendario. La última opción es una asociación al calendario de la plataforma donde haya sido exportado. Además esta última posibilidad permite ver el nombre de todas las tareas que han subido sus profesores hasta ese momento. Para acceder al calendario, que se observa en la Figura 2, los alumnos deben acceder a través del Campus Virtual.

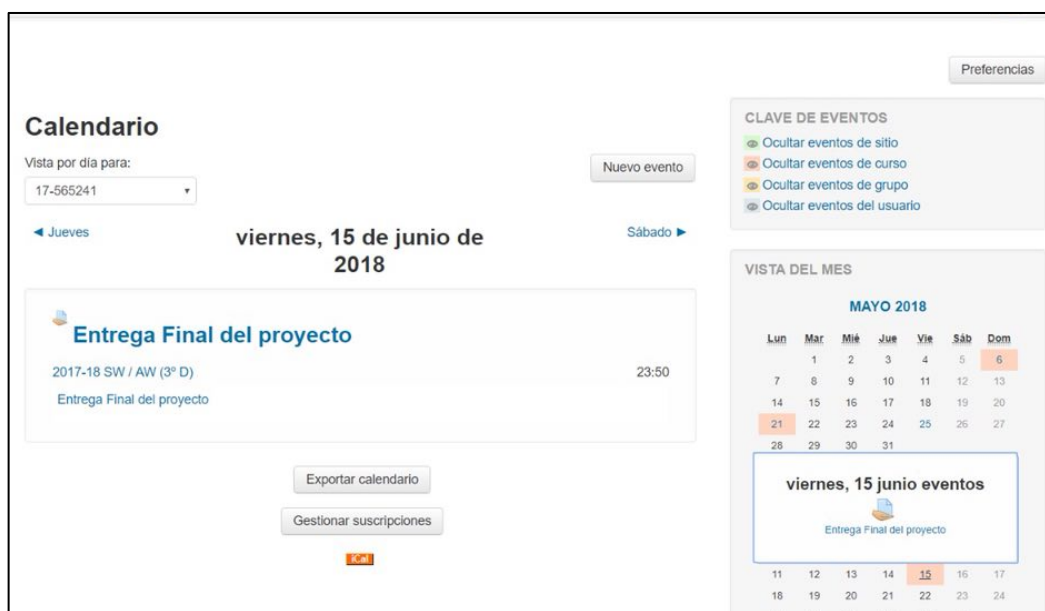


Figura 2: Visualización de una tarea en Calendario

Esta aplicación soluciona un problema importante: el alumno podrá observar cuando tiene que entregar cada una de sus tareas, como por ejemplo los ejercicios que tiene que presentar este mes. Sin embargo, no ofrece la posibilidad de obtener información detallada de cada tarea, ni la información de la asignatura o del profesor que la imparte, ni ver sus respectivas calificaciones. Otro inconveniente es que no está disponible para dispositivos móviles, lo que requiere que el alumno tenga que utilizar el navegador para ver la información deseada. Dicha tarea es tediosa, lo que provoca que los alumnos casi nunca utilicen esta aplicación.

2.3 UCMovil

Es una aplicación para dispositivos móviles que proporciona la información relevante sobre la universidad. Ofrece las siguientes posibilidades [2]:

1. *Consultar información sobre los cursos que son impartidos en la universidad.*
2. *Consultar datos de contacto, horarios y localización.* Tanto del profesorado como de los centros.
3. *Acceder al catalogo de la biblioteca.* Así como acceder a un chat con los bibliotecarios para resolver dudas sobre cada una de las bibliotecas de la universidad.
4. *Proporcionar información sobre las novedades y noticias de la universidad.* Tales como pueden ser conferencias, noticias científicas y concursos.

La aplicación, pese a estar diseñada para dispositivos móviles, no soluciona el problema planteado. Pese a ofrecer la posibilidad de consultar la información del curso del alumno y los datos de contacto de la facultad y de su profesorado. No ofrece información sobre las entregas que poseerán los alumnos durante el curso académico.

2.4 Requisitos de la aplicación y comparativa

Tras el análisis de las aplicaciones utilizadas por los estudiantes, se ha llegado a la conclusión que lo mejor es tener una aplicación que ellos puedan consultar en cualquier momento, de manera optima y precisa. Se ha optado por una aplicación para dispositivos móviles. Al llegar a esta conclusión se plantea otra cuestión: ¿cuál es la plataforma en la que se va a basar la aplicación?

Las plataformas más extendidas son iOS (Apple) y Android (Google), aunque también existen otras plataformas, por ejemplo BlackBerry. Para abordar la cuestión, se plantean las dos formas de desarrollo [3]:

1. *Desarrollo nativo.* Consiste en desarrollar una aplicación para una plataforma en concreto, utilizando su lenguaje nativo. Las aplicaciones que han sido desarrolladas de forma nativa tendrán acceso a características avanzadas del terminal. Su desarrollo es más complejo. Un inconveniente a tener en cuenta, es que su código no es reutilizable para otras plataformas.
2. *Desarrollo multiplataforma o híbrido.* Consiste en desarrollar una aplicación combinando tecnologías de las aplicaciones nativas, que serán un conjunto de API (*Application Programming Interface*) nativas de cada plataforma y de las aplicaciones web. Este tipo de desarrollo permite la reutilización de código para todas las plataformas. También su coste es menor que el desarrollo nativo, ya que no posee acceso a las características avanzadas del terminal. Por lo que ofrecen una peor experiencia al usuario.

En la actualidad la mayoría de los estudiantes poseen móviles con sistema operativo Android, pero hay un gran número de alumnos que poseen móviles con sistema operativo iOS. Teniendo en cuenta dicha información y el estudio de ambos desarrollos, se ha optado por escoger el desarrollo multiplataforma.

Además, tras finalizar el estudio, se ha determinado que la aplicación que utilizan los alumnos debe mostrar una lista con todas las tareas del curso académico. Teniendo en cuenta que el objetivo es ayudar a la organización del alumno, las tareas se ordenan por fecha de entrega. Para ayudar al usuario, se muestra un elemento que indica la prioridad para realizarla, y se muestra la información relevante de dicha tarea.

Aunque no esté relacionado con la organización de las tareas, la aplicación contará con la posibilidad de consultar los datos. Serán los del alumno, las asignaturas a las que asiste y los profesores que imparten dichas asignaturas. También se añadirá la recepción de la calificación de cada entrega. Se ofrece la posibilidad de ver todas las notas de una determinada asignatura para tener un control sobre su progreso académico.

Tras enumerar los aspectos más destacables que tendrá que poseer la aplicación, se muestra en la Tabla 1, una comparación de las aplicaciones existentes y de los aspectos anteriormente mencionados.

	<i>Campus Virtual</i>	<i>Calendario</i>	<i>UCMovil</i>	<i>Requisitos de aplicación a realizar</i>
<i>Dispositivos móviles</i>	X	X	✓	✓
<i>Gestión de base de datos</i>	✓	X	X	X
<i>Consulta datos</i>	✓	X	✓	✓
<i>Avisos y organización</i>	X	X	X	✓
<i>Ver información de una tarea</i>	✓	X	X	✓
<i>Ver todas las tareas</i>	X	✓	X	✓
<i>Ver evaluación de tareas</i>	✓	X	X	✓
<i>Ver calificaciones asignatura</i>	✓	X	X	✓

Tabla 1: Comparativa entre aplicaciones existentes y requisitos de la aplicación

Cuando se hace mención a la aplicación a realizar, se hace referencia a la aplicación que utilizarán los alumnos. Aunque en la Tabla 1 se muestre que no se realizará una gestión en la base de datos, ese punto no es del todo correcto. La aplicación a realizar sí posee una gestión sobre la base de datos, pero se gestiona por el administrador y por el profesorado.

3. ARQUITECTURA DE LA APLICACIÓN

En esta sección se detallarán los aspectos técnicos de la aplicación que se han tenido en cuenta para la realización de este proyecto. Inicialmente se explicará la arquitectura de la aplicación, así como los datos considerados y las consideraciones de implementación en cuanto a seguridad. Finalmente, se comentarán los lenguajes de programación y herramientas de desarrollo utilizados en el proyecto.

3.1 Arquitectura cliente-servidor

El proyecto adopta una arquitectura cliente-servidor a fin de facilitar la distribución de contenidos y reducir los requisitos computacionales de la aplicación. A continuación se explica el papel ejercido por cada uno en la creación del proyecto.

Se ha desarrollado un protocolo REST (*Representational State Transfer*) [4], es decir, el servidor y el cliente son independientes entre sí, y por tanto se separa la interfaz de usuario y el almacenamiento de datos. Este estilo de trabajo es muy utilizado en desarrollo de aplicaciones y de páginas web. Durante la implementación, se aplican las acciones propias de REST, en concreto, se utilizan GET y POST [5]. Este protocolo permite migrar a otros servidores o realizar cambios en la base de datos, siempre y cuando se envíen de forma correcta.

Se entiende por servidor a un proceso que brinda servicio a los clientes que lo soliciten. El servidor se encarga de interpretar y de enviar las respuestas a las solicitudes que han sido realizadas por los clientes. Se cuenta con un único servidor central para el desarrollo de la aplicación.

Se entiende por cliente al equipo que es utilizado por el usuario final y que requiere de los servicios del servidor [6]. Cada usuario utiliza un cliente, se obtiene un total de tres clientes. Son clientes ligeros, es decir, dependen primariamente del servidor para las tareas de procesamiento y se enfocan en transportar la entrada y la salida entre el usuario y el servidor remoto. Este tipo de cliente necesita una conexión estable al servidor remoto, es decir, los clientes necesitaran acceso a Internet. Para las aplicaciones destinadas a administrador y al profesorado, se desarrolla una página web, mientras que para la aplicación destinada a los estudiantes se utiliza una aplicación para dispositivos móviles.

En los siguientes apartados se explicará en que situaciones el servidor y los distintos clientes se comunican.

3.2 Módulos del sistema

Para el desarrollo de la aplicación se ha utilizado el Modelo-Vista-Controlador (MVC) [7]. Este patrón permite separar la lógica de la aplicación de la lógica de la vista. El modelo se encarga de los datos, consultando la base de datos. El controlador recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y comunicárselos a la vista. Las vistas se encargan de recibir las peticiones del usuario y se las trasladan al Controlador.

3.2.1 Diagrama de clases

El diagrama de clases sirve para visualizar las relaciones entre las clases que se encuentran en el sistema. En la Figura 3 se observa el diagrama del sistema. A continuación se explican las clases que lo componen.

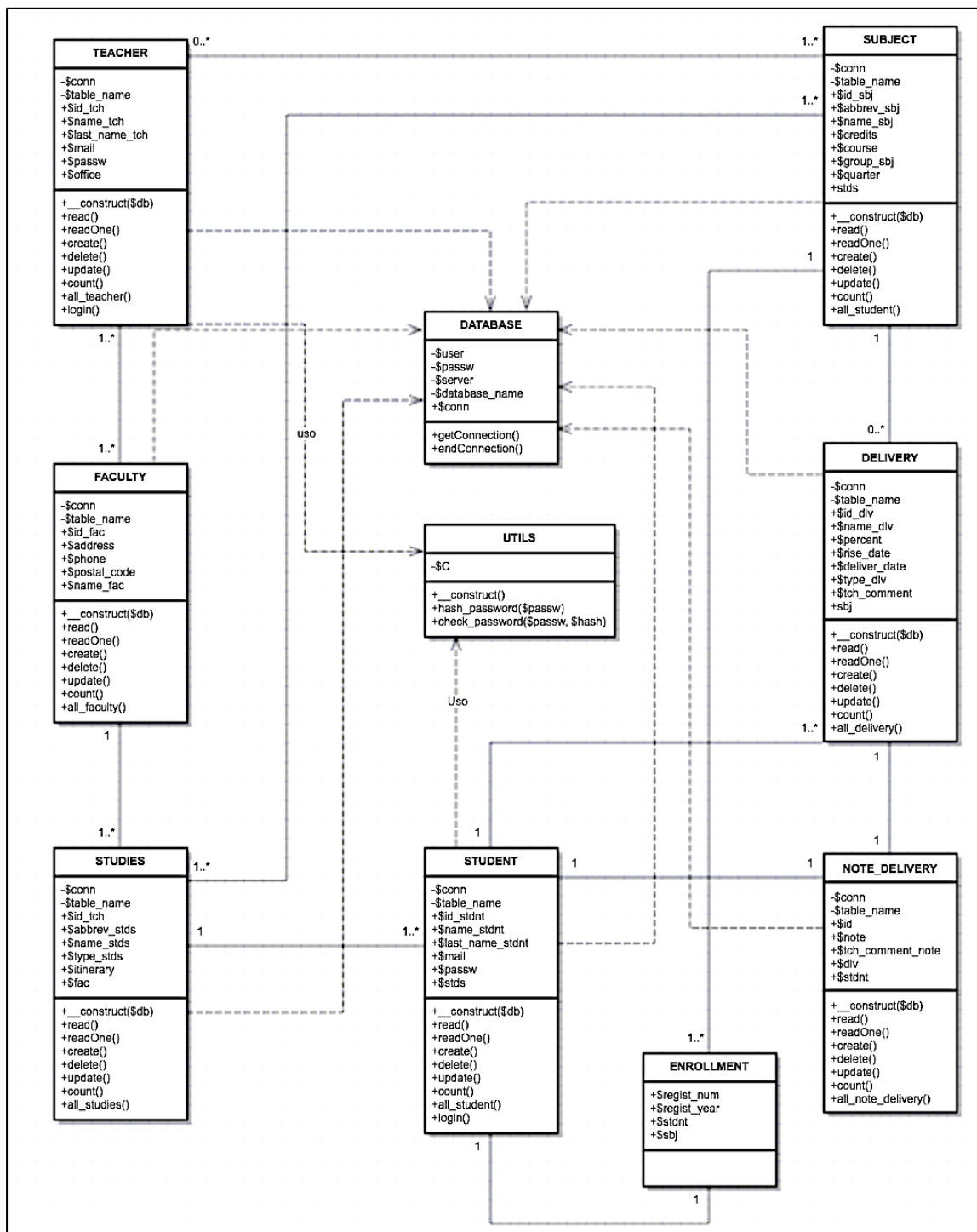


Figura 3: Diagrama de clases

La clase “Database” establece una conexión con la base de datos. Todas las clases, salvo “Utils” y “Enrollment”, poseen un atributo que las relaciona con dicha

clase. Esta clase contiene los atributos necesarios para realizar la conexión y dos métodos. El método “*getConnection()*” que realiza la conexión con la base de datos. El método “*endConnection()*” que cierra dicha conexión.

La clase “*Utils*” es utilizada para realizar el *hashing* de la contraseña, se utiliza en las clases que determinan los usuarios que pueden utilizar la aplicación del proyecto. Por “*Teacher*” que representa al profesorado y al administrador que usan las aplicaciones web, y por “*Student*” que representa al alumnado que utiliza la aplicación para dispositivos móviles. Ambas clases poseen el método “*login()*”.

La clase “*Enrollment*” no posee métodos, se utiliza para obtener la información de matriculación de un estudiante con respecto a una asignatura. Se establece mediante los atributos “*regist_num*” y “*regist_year*”, que sirve para saber el número de veces que se ha matriculado un alumno en dicha asignatura y el año de su primera matriculación.

El resto de clases que aparece en el diagrama poseen los mismos métodos. En la Tabla 2 se observan esos métodos y en que consiste cada uno de ellos.

Método	Descripción
<i>read()</i>	Lista de elementos
<i>readOne()</i>	Información de un elemento concreto
<i>create()</i>	Crea un nuevo elemento
<i>delete()</i>	Elimina un elemento concreto
<i>update()</i>	Edita información de un elemento concreto
<i>count()</i>	Cuenta el número de elementos
<i>all_clase()</i>	Obtiene elementos en función de otro

Tabla 2: Métodos comunes de clases

Además, se tiene en cuenta el constructor que las relaciona con la clase “*Database*”, para así poder realizar la conexión con la base de datos. Para poder realizar dicha conexión, las clases deben contar con dos atributos. El atributo “*conn*” y atributo “*table_name*”, se relacionan con la clase “*Database*” y el nombre de la tabla de la base de datos, respectivamente.

3.2.2 Diagrama de secuencia

Un diagrama de secuencia muestra el desarrollo de una interacción en base al intercambio de mensajes realizados entre el cliente y el servidor. Para modelar el comportamiento del sistema se han realizado los diagramas en función del tipo de acción a realizar, teniendo en cuenta el MVC. A continuación se detallan dos casos que dependen de la acción a realizar.

En el caso de “*Login*” o “*Editar contraseña*”, los objetos que intercambian mensajes son los mismos. El objeto “*Usuario*” que representa al administrador, profesor o estudiante, es el encargado de realizar la acción vinculada al usuario que representa, como por ejemplo, pulsar un botón o rellenar un formulario. El objeto “*Application*” representa a las distintas aplicaciones del proyecto, es decir, a los clientes y se establece como la vista del patrón. La clase “*User*” que representa a las clases de los usuarios, las cuales serían “*Teacher*” y “*Student*”, se establece como el controlador del patrón, que representaría al servidor del sistema. Por lo tanto, la comunicación entre el cliente y el

servidor se establece en el intercambio de mensajes entre “User” y “Application”. Y finalmente, “Database” y “BD” que representan la relación entre el sistema y la base de datos, se establece como el modelo.

En la Figura 4, se pueden observar los mensajes intercambiados para la realización de *Login*. Los cuales se resumen:

1. La vista se encuentra en la interfaz login.
2. El usuario rellena el formulario.
3. El usuario envía los datos que ha rellenado.
4. El cliente se comunica con el servidor mediante la operación “login()”.
5. El servidor establece una conexión con la base de datos.
6. La base de datos busca la información enviada.
7. La base de datos realiza una comprobación de dichos datos.
8. La base de datos envía un mensaje de confirmación o de error.
9. Si dicho mensaje es de confirmación, se envían los datos. En caso de error, se envía un mensaje indicando que se ha producido un error.
10. El servidor comunica al cliente la información sobre dicho mensaje.
11. Se establece el cierre de conexión de la base de datos.
12. En caso de éxito, la vista cambia. Si no, se mantiene mostrando el mensaje de error.

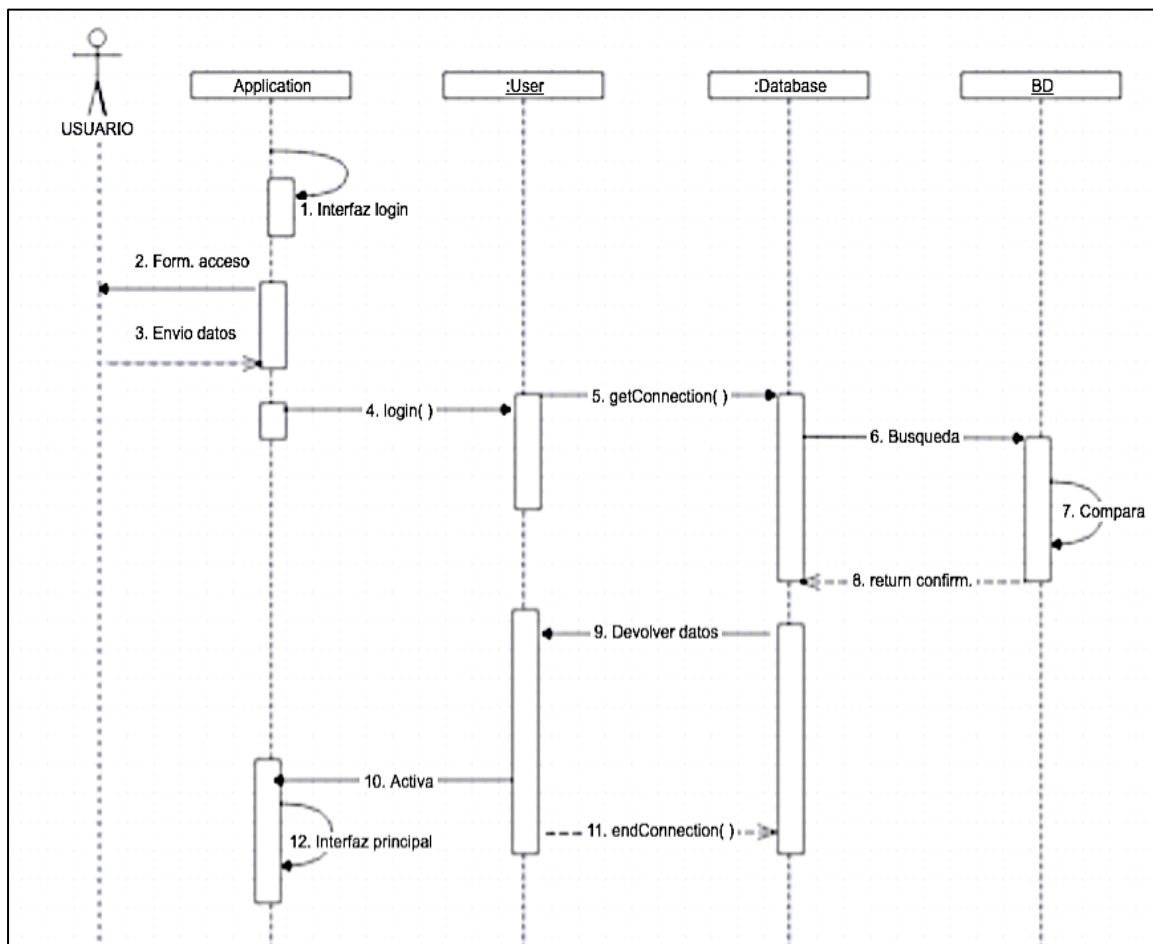


Figura 4: Diagrama secuencia – Login

A continuación, en la Figura 5, se pueden observar los mensajes intercambiados para la realización de “*Editar contraseña*”. Cabe destacar que el objeto “Utils”

representa a la clase que realiza el *hashing* de la contraseña. El *hashing* se explica en el apartado 3.3 de la memoria. Los mensajes intercambiados para la realización de “*Editar contraseña*” se resumen:

1. La vista se encuentra en la interfaz de editar la contraseña.
2. El usuario rellena el formulario.
3. El usuario envía los datos que ha rellenado.
4. El cliente se comunica con el servidor mediante “*readOne()*”
5. El servidor establece una conexión con la base de datos.
6. La base de datos busca la información enviada.
7. La base de datos realiza una comprobación de dichos datos.
8. La base de datos envía un mensaje de confirmación o de error.
9. Si dicho mensaje es de confirmación, se envían los datos. En caso contrario se envía un mensaje indicando que se ha producido un error.
10. El servidor comunica al cliente la información sobre dicho mensaje.
11. Se establece el cierre de conexión de la base de datos.
12. En caso de error se muestra el mensaje de error. Si se ha producido un éxito, se realiza una comprobación de la contraseña antigua con el hash generado.
13. La clase encargada del *hashing* devuelve la respuesta.
14. En caso de error se muestra el mensaje de error. Si se ha producido un éxito, se realiza la comprobación de la nueva contraseña.
15. Si la comprobación ha sido un éxito, se realiza el *hashing* de la nueva contraseña.
16. La clase encargada del *hashing* devuelve la respuesta.
17. En caso de error, se muestra un mensaje de error. En caso contrario, el cliente se comunica con el servidor mediante “*update()*”.
18. El servidor establece una conexión con la base de datos.
19. La base de datos edita la información en función de la nueva contraseña.
20. La base de datos envía un mensaje de confirmación o de error.
21. Si dicho mensaje es de confirmación, se envían los datos. En caso contrario se envía un mensaje indicando que se ha producido un error.
22. El servidor comunica al cliente la información sobre dicho mensaje.
23. Se establece el cierre de conexión de la base de datos.
24. En caso de error se muestra el mensaje de error. Si se ha producido un éxito, se muestra el mensaje de confirmación.

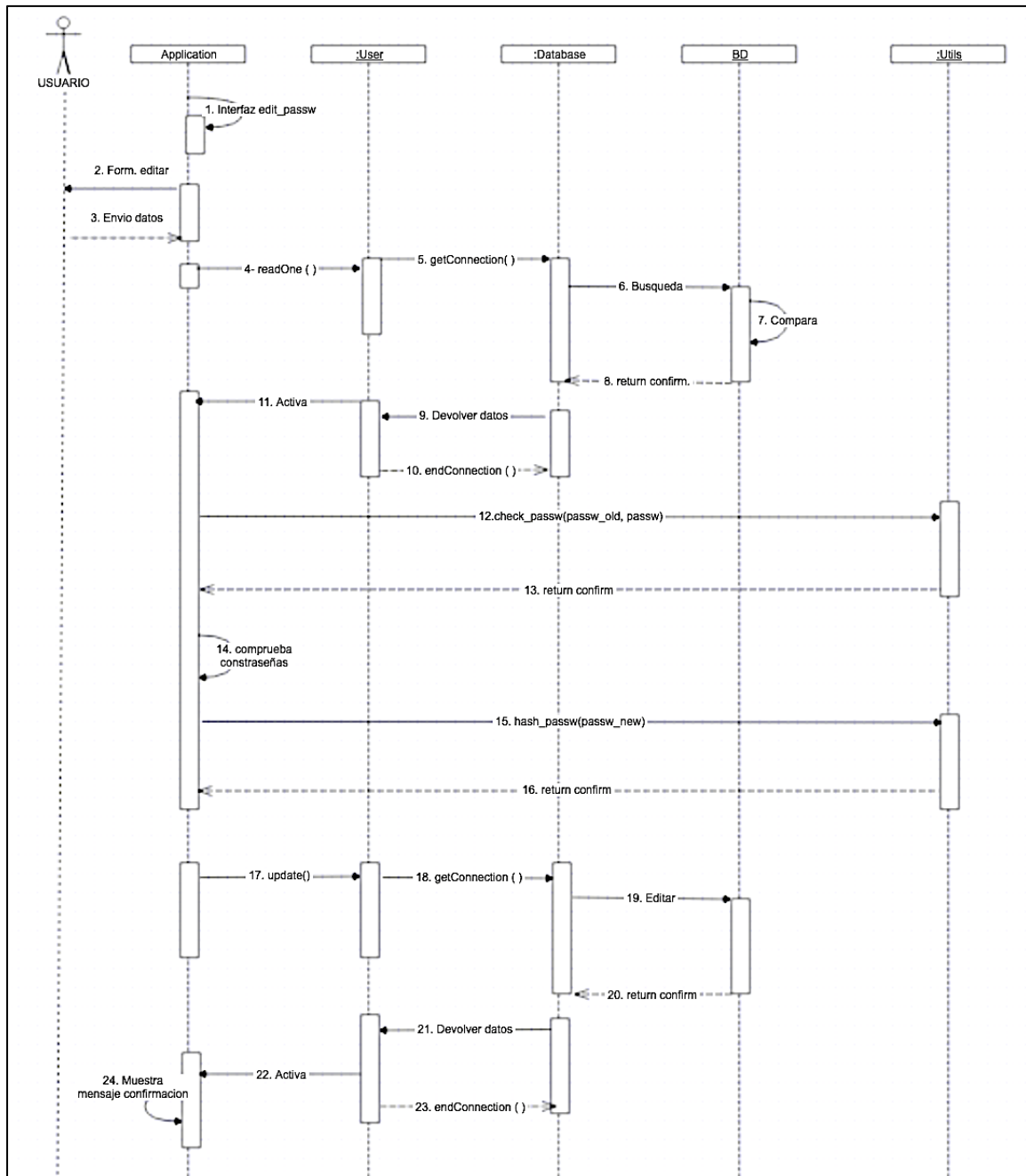


Figura 5: Diagrama secuencia - Editar contraseña

Finalmente, en caso de que la acción a realizar sea “Ver”, “Ver información”, “Editar”, “Crear” o “Eliminar”, los objetos que intercambian mensajes serán los mismos. El objeto “Usuario” que representa al administrador, profesor o estudiante, es el encargado de realizar la acción, como por ejemplo, pulsar un botón o rellenar un formulario. El objeto “Application” representa a las distintas aplicaciones del proyecto, es decir, a los clientes y se establece como la vista del patrón. La clase “Object” se establece como el controlador del patrón, que representaría al servidor del sistema. Dicha clase se corresponde con cualquier elemento de la tabla de la base de datos, como por ejemplo, un estudiante, una asignatura, una tarea... Como se está expresando de forma genérica no se establecen las diferencias entre las distintas acciones o contenidos para cada usuario, Por lo tanto, la comunicación entre el cliente y el servidor se establece en el intercambio de mensajes entre “Object” y “Application”. “Database” y “BD” que

representan la relación entre el sistema y la base de datos, se establece como el modelo del MVC.

En la Figura 6, se pueden observar los mensajes intercambiados para la realización de “*Ver todo*”. Se resumen:

1. La vista se encuentra en la interfaz de la página principal.
2. En la vista se encuentra el botón “*Ver*”.
3. El usuario pulsa el botón.
4. El cliente se comunica con el servidor mediante la operación “*read()*”.
5. El servidor establece una conexión con la base de datos.
6. La base de datos busca la información enviada.
7. La base de datos envía un mensaje de confirmación o de error.
8. Si dicho mensaje es de confirmación, se envían los datos. En caso contrario se envía un mensaje indicando que se ha producido un error.
9. El servidor comunica al cliente la información sobre dicho mensaje.
10. Se establece el cierre de conexión de la base de datos.
11. En caso de éxito, la vista muestra la lista de datos. Si no, muestra el mensaje de error.

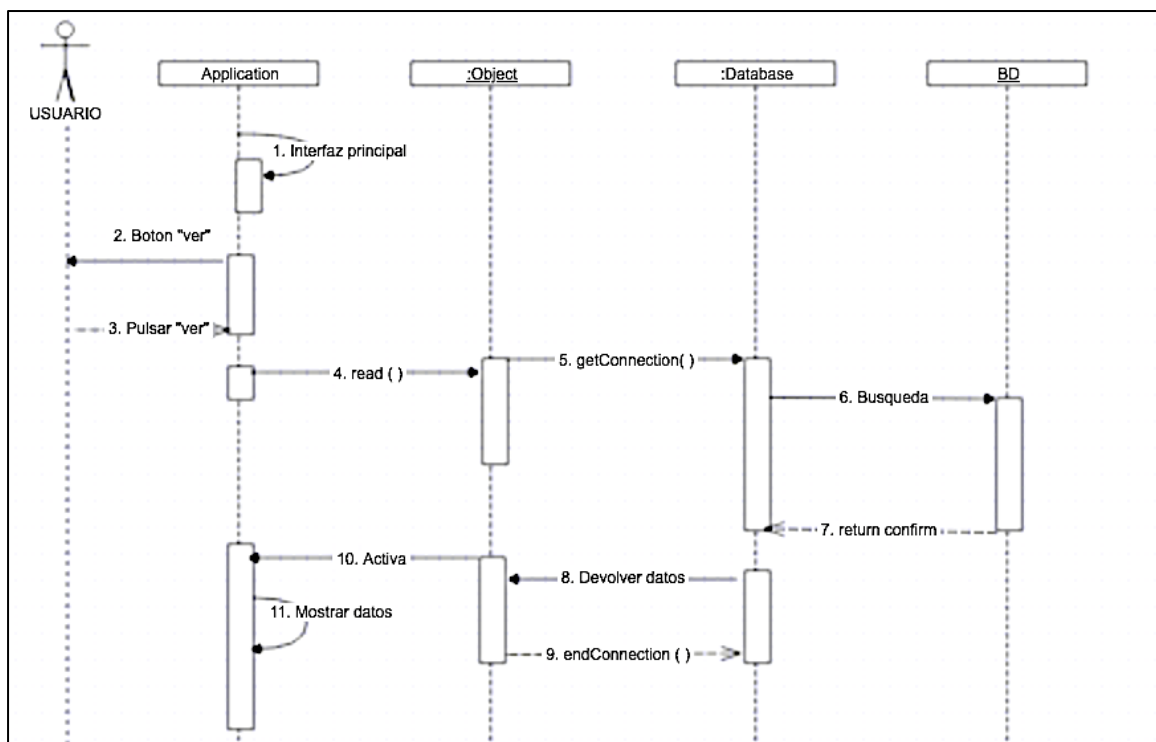


Figura 6: Diagrama secuencia - Ver toda la información

En la Figura 7, se pueden observar los mensajes intercambiados para la realización de “*Ver información concreta*”, que se comportan de la siguiente manera:

1. La vista se encuentra en la interfaz de la página principal.
2. En la vista se encuentra el botón “*Info*”.
3. El usuario pulsa el botón.
4. El cliente se comunica con el servidor mediante la operación “*readOne()*”.
5. El servidor establece una conexión con la base de datos.
6. La base de datos busca la información enviada.
7. La base de datos compara el identificador de la tabla.

8. La base de datos envía un mensaje de confirmación o de error.
9. Si dicho mensaje es de confirmación, se envían los datos. En caso contrario se envía un mensaje indicando que se ha producido un error.
10. El servidor comunica al cliente la información sobre dicho mensaje.
11. Se establece el cierre de conexión de la base de datos.
12. En caso de éxito, la vista muestra la información del dato concreto. Si no, muestra el mensaje de error.

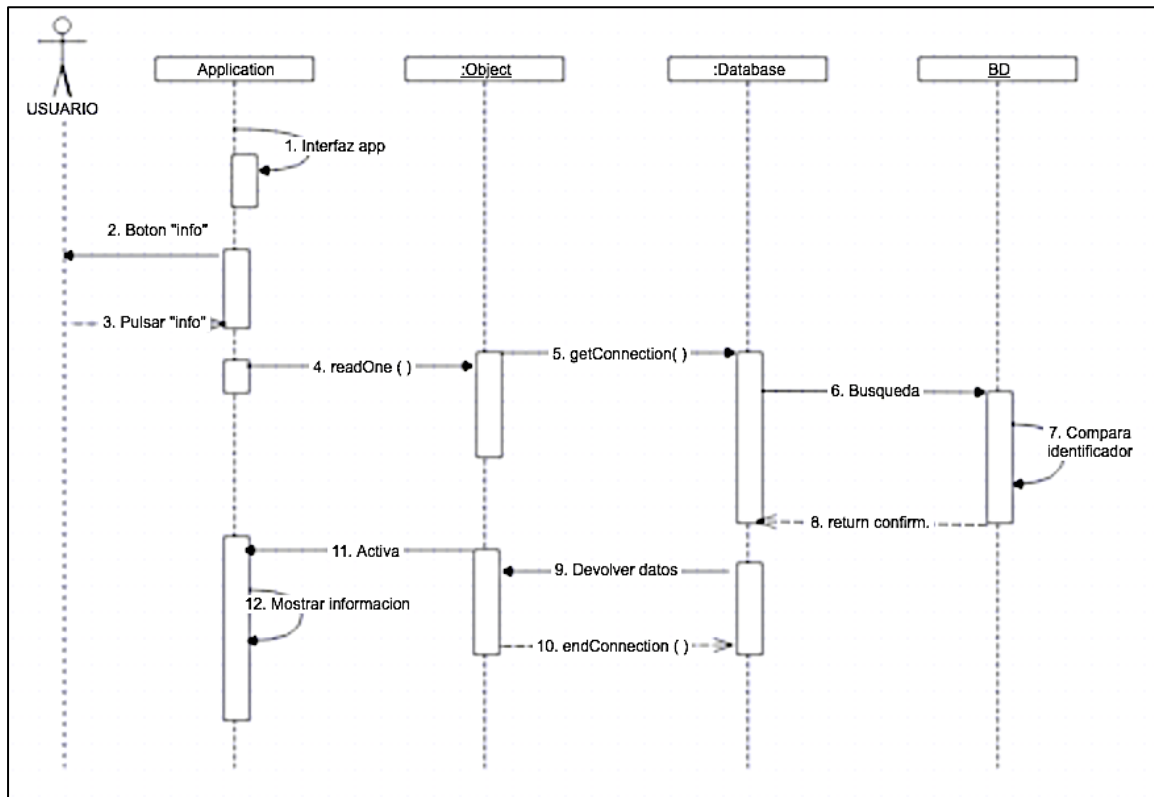


Figura 7: Diagrama secuencia - Ver información de un dato concreto

En la Figura 8, se pueden observar los mensajes intercambiados para la realización de "Editar". Se resumen:

1. La vista se encuentra en la interfaz de editar un dato.
2. El usuario rellena el formulario.
3. El usuario envía los datos del formulario.
4. El cliente se comunica con el servidor mediante la operación "update()".
5. El servidor establece una conexión con la base de datos.
6. La base de datos busca la información enviada.
7. La base de datos compara el identificador de la tabla.
8. La base de datos envía un mensaje de confirmación o de error.
9. Si dicho mensaje es de confirmación, se envían los datos. En caso contrario se envía un mensaje indicando que se ha producido un error.
10. El servidor comunica al cliente la información sobre dicho mensaje.
11. Se establece el cierre de conexión de la base de datos.
12. En caso de éxito, la vista muestra mensaje de confirmación. Si no, muestra el mensaje de error.

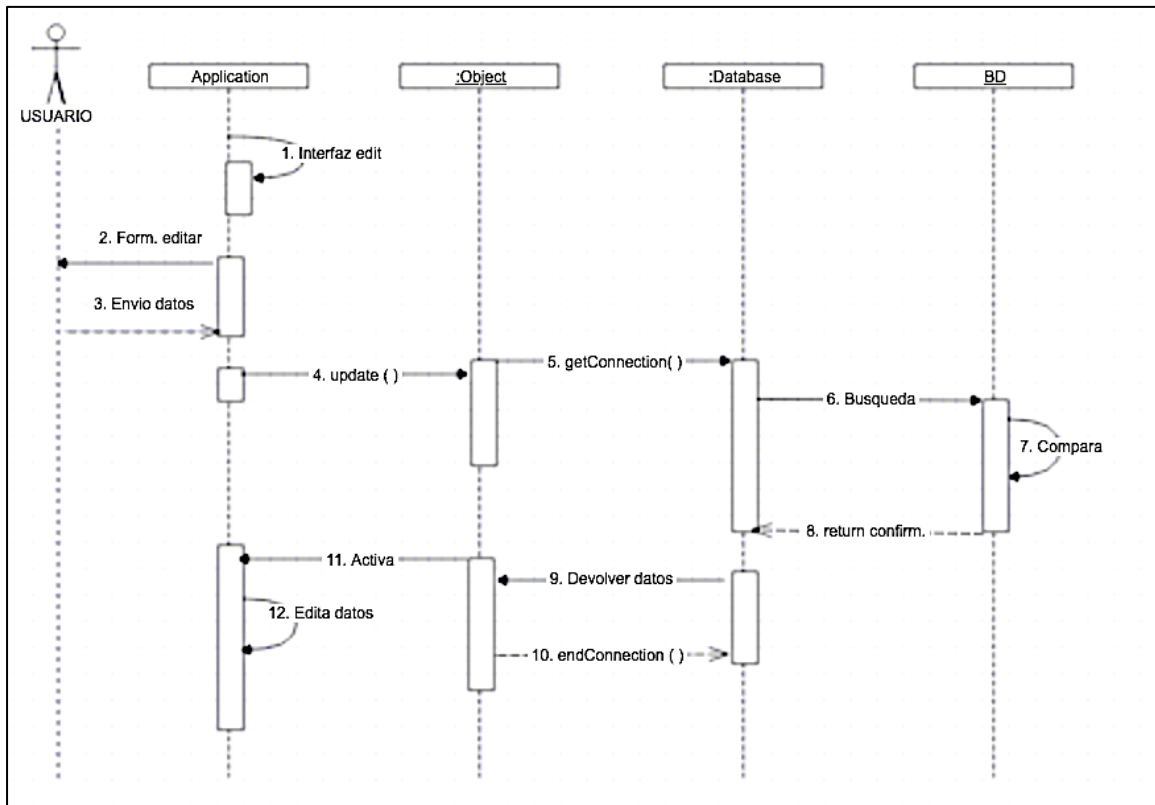


Figura 8: Diagrama secuencia - Editar dato

En la Figura 9, se pueden observar los mensajes intercambiados para la realización de “Eliminar”. Se resumen:

1. La vista se encuentra en la interfaz de la página principal.
2. En la vista se encuentra el botón “Borrar”.
3. El usuario pulsa el botón.
4. El cliente se comunica con el servidor mediante la operación “delete()”.
5. El servidor establece una conexión con la base de datos.
6. La base de datos busca la información enviada.
7. La base de datos compara el identificador de la tabla.
8. La base de datos envía un mensaje de confirmación o de error.
9. Si dicho mensaje es de confirmación, se envían los datos. En caso contrario se envía un mensaje indicando que se ha producido un error.
10. El servidor comunica al cliente la información sobre dicho mensaje.
11. Se establece el cierre de conexión de la base de datos.
12. En caso de éxito, la vista muestra mensaje de confirmación. Si no, muestra el mensaje de error.

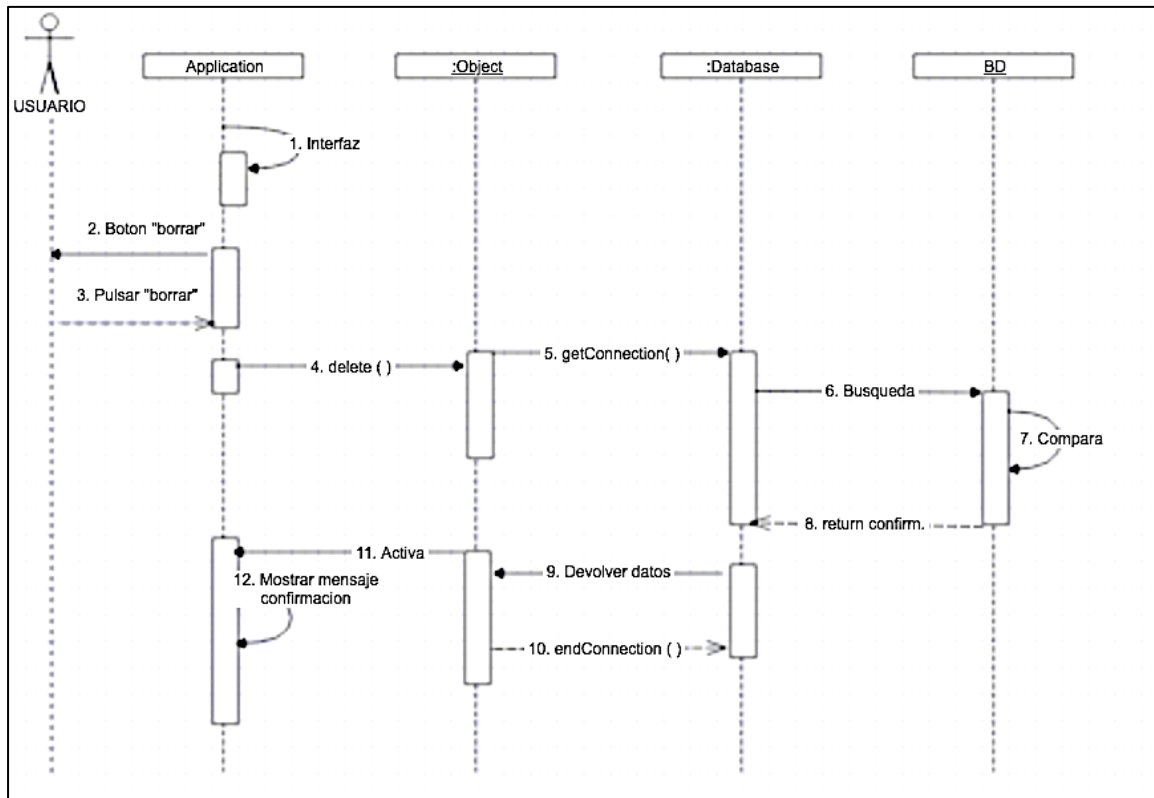


Figura 9: Diagrama secuencia - Eliminar dato

Terminando con los diagramas de secuencia, en la Figura 10, se pueden observar los mensajes intercambiados para la realización de “*Crear*”. Se establecen de manera:

1. La vista se encuentra en la interfaz de crear.
2. En la vista se encuentra el botón “*Crear*”.
3. El usuario pulsa el botón.
4. El cliente se comunica con el servidor mediante la operación “*create()*”.
5. El servidor establece una conexión con la base de datos.
6. La base de datos busca la información enviada.
7. La base de datos compara el identificador de la tabla.
8. La base de datos envía un mensaje de confirmación o de error.
9. Si dicho mensaje es de confirmación, se envían los datos. En caso contrario se envía un mensaje indicando que se ha producido un error.
10. El servidor comunica al cliente la información sobre dicho mensaje.
11. Se cierra la conexión con la base de datos.
12. En caso de éxito, la vista muestra mensaje de confirmación. Si no, muestra el mensaje de error.

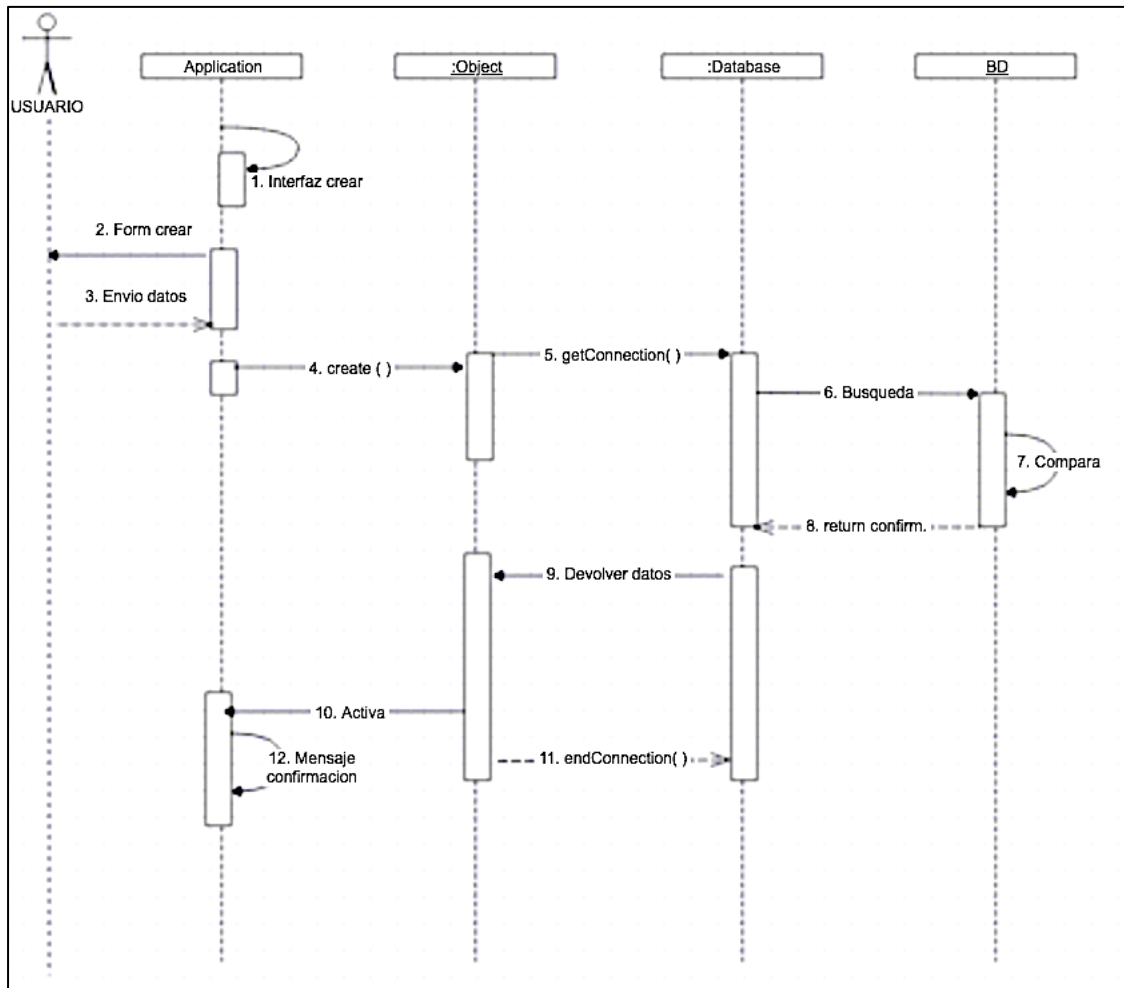


Figura 10: Diagrama secuencia - Crear dato

3.3 Consideraciones de implementación

Se ha diseñado un sistema en donde cada usuario sólo podrá ver la parte de la aplicación adecuada a su perfil. Esto se traduce en que a un administrador se le permite realizar cambios en los elementos de la base de datos que representen una información, como puede ser una facultad, un estudiante, etcétera. A un profesor se le permite realizar cambios en la base de datos, pero sólo sobre las tareas de las asignaturas que imparte, incluyendo la calificación de estas. A un estudiante se le otorga el derecho de ver la información de las asignaturas en las que está matriculado, incluyen las tareas de dicha asignatura y el profesor que la imparte, pero no se le permite el acceso a la base de datos.

Se han establecido algunos supuestos y dependencias a la hora de desarrollar la aplicación. Se considera un supuesto a un hecho que se cumple en el desarrollo. Se ha establecido que todo usuario tiene un correo y una contraseña únicos y que siempre se produce un proceso de login antes de usar la aplicación. Se considera una dependencia a un factor que puede afectar a los requisitos del sistema, se ha tenido en cuenta que se pueden llegar a producir problemas con la base de datos o con el acceso a Internet; por lo que es importante que ambas sean seguras ya que se pueden producir pérdidas de datos si hay problemas de conexión, un ejemplo es a la hora de dar de alta a un nuevo alumno o una nueva tarea.

Para el correcto desarrollo de la aplicación se tiene en cuenta los problemas que pueden surgir a nivel de seguridad. Se realizan comprobaciones en función de la existencia de un elemento, un ejemplo es antes de insertar un nuevo elemento se debe comprobar que no existe en la base de datos actual, o antes de editar la información de un elemento, se comprueba que dicho elemento existe. También se realizan comprobaciones según el tipo de elemento o de la inserción correcta de datos a la hora de realizar cambios en la base de datos. Para asegurar la criptografía de la contraseña de un usuario, en la base de datos aparecerá un *hash* de dicha contraseña [8]. Finalmente, en cuanto a nivel de seguridad, se ha establecido un *token* para cada usuario, exactamente un JWT (*Json Web Token*) [9] que es un conjunto de medios de seguridad para peticiones HTTP (*HiperText Transfer Protocol*). JWT se almacena en el lado del cliente, donde cada petición que realiza el usuario va acompañada de un *token*, que permite al API identificar al usuario y redirigir el flujo de la aplicación.

3.4 Lenguajes y herramientas de desarrollo

Se ha montado un servidor local para poder realizar las pruebas durante la implementación de la aplicación. Para lograrlo se ha utilizado la aplicación MAMP [10], que es la versión de XAMP para iOS, que facilita Apache, MySQL [11] y PHP [12].

Al utilizar un protocolo REST [4], para el desarrollo del servidor se desarrolla con PHP. Se trata de un lenguaje de código abierto ampliamente extendido para la creación de páginas web al lado del servidor, y que facilita el acoplamiento a la base de datos MySQL [11]. Para la transmisión de datos se ha utilizado JSON (*JavaScript Object Notation*) que es un formato de texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript. Los JSON cadenas que se utilizan para transmitir datos a través de la red y que se convierten en un objeto nativo de JavaScript cuando se quiere acceder a los datos. Finalmente, se han comprobado dichos datos mediante la aplicación *Advanced REST client* que es una extensión de Chrome que permite lanzar peticiones a APIs RestFul, que es el tipo de servidor que utiliza la aplicación, como se ha mencionado en el apartado 3.1 de la memoria.

Para el desarrollo del cliente se han utilizado distintos tipos de lenguaje. Como se ha mencionado en el punto 2.4 de la memoria, se desea realizar un desarrollo multiplataforma en la aplicación diseñada para los estudiantes, es decir, para la aplicación utilizada por dispositivos móviles. Para conseguir dicho desarrollado, se utiliza el framework de Apache Cordova [13], que es un marco de desarrollo móvil de código abierto para el desarrollo multiplataforma. Se encarga de interpretar, compilar y generar la aplicación nativa a partir de tecnologías web. Los lenguajes que se pueden utilizar en este framework son HTML5 (*HyperText Markup Language*) [14], CSS3 (*Cascading Style Sheets*) [15] y JavaScript. HTML5 y CSS3 se utilizan para crear el diseño de la interfaz. Ambos se utilizan normalmente en el desarrollo web para dar estilo y presentar las vistas de la aplicación a desarrollar. JavaScript se encarga de implementar la funcionalidad de la aplicación, utilizando AJAX (*Asynchronous JavaScript and XML*) [16]. AJAX se utiliza para la comunicación entre el servidor y los distintos cliente, ya que se usa para enviar las peticiones al servidor que enviará la respuesta en un JavaScript.

Para conseguir el nivel de seguridad utilizando JWT es necesario utilizar lenguajes que permitan la recepción de peticiones HTTP. Para el desarrollo web del

cliente se ha utilizado JavaScript y AJAX que reciben las peticiones mediante JSON. HTML5, CSS3, y Bootstrap [17], son utilizados para crear el diseño de la interfaz. Este último es un framework de código abierto para diseño de sitios y aplicaciones web, contiene plantillas de diseño basado en HTML y CSS. Además, solo se ocupa del desarrollo front-end, es decir, de la parte del software que interactúa con los usuarios, lo que facilita la creación del diseño de la aplicación.

4. MODELO DE DATOS

Para el almacenamiento de datos de las aplicaciones se ha decidido utilizar una base de datos relacional que contiene toda la información sobre los usuarios y las tareas. Durante la realización de la implementación se ha utilizado phpMyAdmin [18], que es una herramienta que maneja la administración de MySQL a través de páginas web.

La aplicación se desarrolla para el uso de los estudiantes, representados por *Student*. Hace referencia al alumno que utilizará la aplicación y se compone de un identificador único y de sus datos personales, entre los que se incluye su correo de la UCM que es único.

La aplicación se crea para ayudar a los estudiantes a organizar sus tareas, representadas por *Delivery*. Hace referencia al conjunto de tareas que puede tener un alumno. Se compone de un identificador único y los atributos necesarios para desarrollar dicha tarea, que se pueden observar en el apartado 3.2.1 de la memoria. Una tarea puede ser de distinto tipo, los cuáles son *prácticas*, *proyectos*, *exámenes* y *ejercicios*, hace referencia al atributo *type_dlv*. Otro atributo a tener en cuenta es *percent* que representa el porcentaje que tendrá una tarea en cuanto a la evaluación global de la asignatura a la que pertenece. Así mismo, se tendrá en cuenta que toda tarea tendrá una fecha de subida, determinada por *rise_date*, y una fecha que represente el día máximo para entregarla o realizarla, determinada por *deliver_date*. Ambas fechas se utilizan para el control de notificaciones y organización del estudiante. Cada entrega pertenece a una asignatura concreta.

Toda tarea dispondrá de una evaluación, representada por *Note_Delivery*. Hace referencia a la nota que le asignará el profesor titular de la asignatura de dicha tarea. Se compone de un identificador único y dos atributos, *note*, que representa la evaluación realizada, y *tch_comment_note*, que constituye el mensaje del profesor sobre dicha evaluación. Este último campo puede ser nulo. Cada nota pertenece a una tarea y a un alumno concreto.

Una tarea pertenece a una asignatura concreta, por lo que se crea la clase *Subject*. Se compone de un identificador único y de varios atributos, entre los que se incluye *quarter*. Este atributo se utiliza para describir el tipo de cuatrimestre al que pertenece una asignatura; los cuales pueden ser *primer*, *segundo* o *anual*. Se asigna un curso y un grupo por cada asignatura. Una asignatura se puede impartir en varios grupos o cursos, para que este último atributo sea válido, la asignatura debe pertenecer a dos tipo de estudio distintos. Cada asignatura pertenece a un tipo de estudio concreto y debe ser impartida por un profesor.

Todo estudiante está matriculado mínimo en una asignatura, por lo que se registra el primer año en que se matriculó de dicha asignatura y el número de veces que la ha cursado.

Una asignatura se imparte por un profesor, representado por *Teacher*. Se compone por un identificador único y sus datos personales, entre los que se incluye su correo de la UCM y el número de su despacho. Todo profesor trabaja mínimo en una facultad, en la que se encuentra su despacho.

Todo estudiante asiste a una facultad, se representa con la clase *Faculty*. Se compone de un identificador único y de la información que la representa, como es su nombre, dirección y teléfono.

En una facultad existen distintos tipos de estudios. Para distinguirlos se crea *Studies*. Un estudio puede ser un *grado*, *master* o *licenciatura*. A su vez, cada uno de ellos puede tener itinerarios. Todo alumno está matriculado en un tipo de estudio, y por tanto, estudia en una facultad.

Para completar la información, en la Figura 11 se puede observar el diagrama entidad-relación que se ha utilizado en este trabajo. Se relaciona con el diagrama de clases mencionado en el apartado 3.2.1 de la memoria.

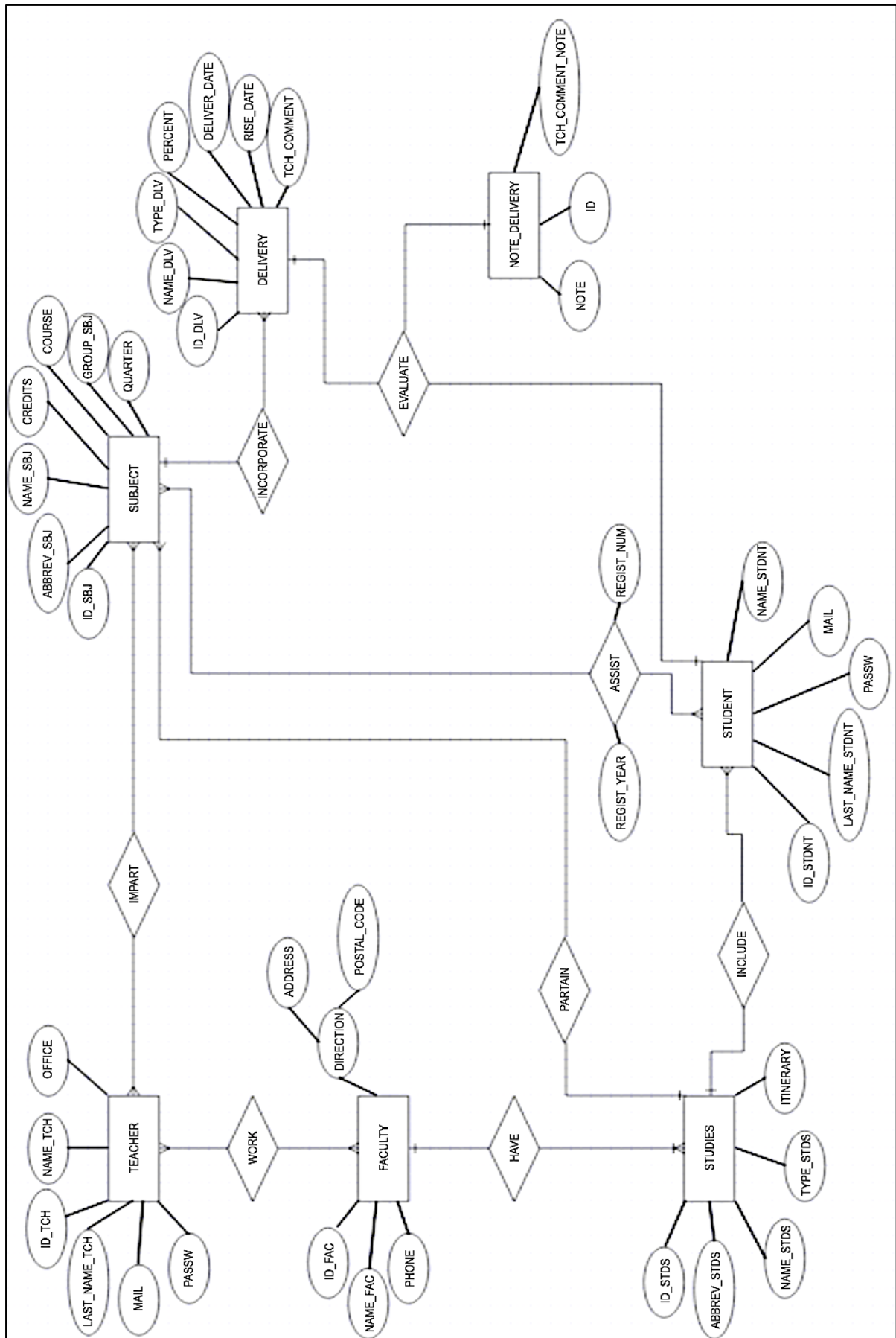


Figura 11: Diagrama entidad-relación

5. DISEÑO INTERFAZ DE USUARIO

Uno de los aspectos más importantes a la hora de realizar una aplicación es la forma de atraer y mantener a los usuarios gracias a su diseño visual. Una aplicación interactúa con el usuario a través de la interfaz de la aplicación. Por ello es muy importante cuidar los aspectos que facilitan dicha interacción y hacen que la experiencia del usuario sea satisfactoria. Para lograr este objetivo se han tenido en cuenta las diez heurísticas de Jakob Nielsen (Nielsen, 1995) [19]. Dichas heurísticas son las siguientes:

1. *Visibilidad del estado del sistema.* Toda aplicación debe mantener siempre informado al usuario de lo que está ocurriendo y brindarle una respuesta en el menor tiempo posible.
2. *Relación entre el sistema y el mundo real.* Toda aplicación debe incluir vocabulario familiar para el usuario. Además, la información que se mostrará debe aparecer en un orden lógico y natural.
3. *Control y libertad de usuario.* El usuario siempre poseerá el control de volver a un estado anterior de la aplicación tras haber cometido un error. Además, de tener la posibilidad de volver a rehacer una acción.
4. *Consistencias y estándares.* Los usuarios de la aplicación deberán saber en todo momento que significan las acciones, signos o palabras mostradas en la aplicación. Para ello hay que mostrar siempre las convenciones establecidas.
5. *Prevención de errores.* Se debe crear un diseño que sea cuidadoso y que ayude al usuario a que no cometa errores, un ejemplo es la funcionalidad de autocompletar.
6. *Reconocer antes de recordar.* Hacer visibles acciones y opciones para que el usuario no tenga que recordar como se realiza una acción.
7. *Flexibilidad y eficiencia de uso.* Toda aplicación debe ser utilizable.
8. *Diseño estético y minimalista.* Se debe evitar cualquier tipo de información innecesaria. Cada información extra compite con la información relevante y disminuye su visibilidad.
9. *Ayuda a los usuarios a reconocer.* Todo mensaje de error deberá ir escrito en un lenguaje sencillo y con instrucciones y soluciones a dicho problema.
10. *Ayuda y documentación.* Toda aplicación debe tener un manual de usuario, aunque solo es imprescindible en aplicaciones complejas.

Teniendo en cuentas las heurísticas mencionadas, se ha realizado un diseño inicial de la aplicación antes de proceder a su implementación. Para realizar el boceto de la aplicación, también denominado “Mockup”, se ha utilizado JustinMind Prototyper [20]. Es una herramienta de creación de prototipos de aplicaciones móviles y web que ofrece capacidades que normalmente se encuentran en herramientas de diagramación y admite la

simulación de alta fidelidad de aplicaciones enriquecidas en internet. A continuación se detalla como las aplicaciones cumplen con las heurísticas mencionadas.

La heurística “*Visibilidad del estado del sistema*” se cumple en ambos diseños, de manera que si se ha producido un error en el sistema, se mostrará un mensaje indicando dicho error.

Para cumplir con “*Relación entre el sistema y el mundo real*”, se utiliza un lenguaje sencillo en la aplicación, de manera que los botones son verbos como ver, crear, editar y borrar. La información que se muestra es precisa, como se observa en la Figura 12 el boceto utilizado en el diseño web. Al usar este funcionamiento, se cumple la heurística “*Consistencias y estándares*”, de manera que el usuario sabe en que consiste la futura acción a realizar.

Home

Exit

LISTA DE PROFESORES

CREAR PROFESOR

NOMBRE	APELLIDO	CORREO	DESPACHO	ACCIONES
n_1	ap_1	c_1	d_1	<div>VER</div> <div>EDITAR</div> <div>BORRAR</div>
n_2	ap_2	c_2	d_2	<div>VER</div> <div>EDITAR</div> <div>BORRAR</div>
n_3	ap_3	c_3	d_3	<div>VER</div> <div>EDITAR</div> <div>BORRAR</div>
n_4	ap_4	c_4	d_4	<div>VER</div> <div>EDITAR</div> <div>BORRAR</div>
n_5	ap_5	c_5	d_5	<div>VER</div> <div>EDITAR</div> <div>BORRAR</div>

«

1

2

3

»

Figura 12: Boceto web - Lista de elementos

Es importante que se cumpla “*Control y libertad de usuario*” y por lo tanto, en cada página siempre se puede volver a la página principal de la aplicación o a la página donde se realizó la acción para llegar a dicha página. Las acciones que determinan un cambio en la base de datos, como son crear, editar y eliminar, poseen la opción de cancelar dichos cambios. Para eliminar un elemento se realiza una pregunta de confirmación. Para crear o editar un elemento es necesario pulsar el botón situado en el formulario de la acción a realizar.

A la hora de introducir datos, la aplicación cuenta con la funcionalidad autocompletar, utilizada por ejemplo en el login que utilizan ambos diseños. Cuando el administrador o profesor crea o edita un elemento, la aplicación cuenta con la opción de seleccionar la opción válida, como se muestra en la Figura 13 el boceto utilizado en el diseño web. Un ejemplo es seleccionar cuál es el tipo de estudio donde está matriculado un alumno. Usando estas funcionalidad, se cumple la heurística “*Prevención de errores*” y “*Ayuda a los usuarios a reconocer*”.

Exit

VER INFORMACIÓN DE UN ALUMNO

LISTA ALUMNOS

NOMBRE	NOMBRE_ALUMNO
APELLIDO	AP_ALUMNO
CORREO	CORREO_AL@UCM.ES
ESTUDIO	GRADO EN ING. INFORMATICA ▼

Figura 13: Boceto web - Ver información de un elemento

Para cumplir con “*Reconocer antes de recordar*”, se ha determinado un diseño de la aplicación dependiendo del tipo de usuario. Si el usuario es el administrador o un profesor, las aplicaciones se muestran de manera muy similar. Ambos contienen los mismo botones de cabecera en la misma posición. Y acciones con una vista parecida, como ver información, editar un dato, crear un dato o mostrar un dato a eliminar, que muestran con una tabla o un formulario la información, como se puede ver la Figura 13.

Si el usuario es un alumno, se muestra un diseño muy similar, al mostrar una lista de elementos y al mostrar la información de un único elemento, como se observa en la Figura 14

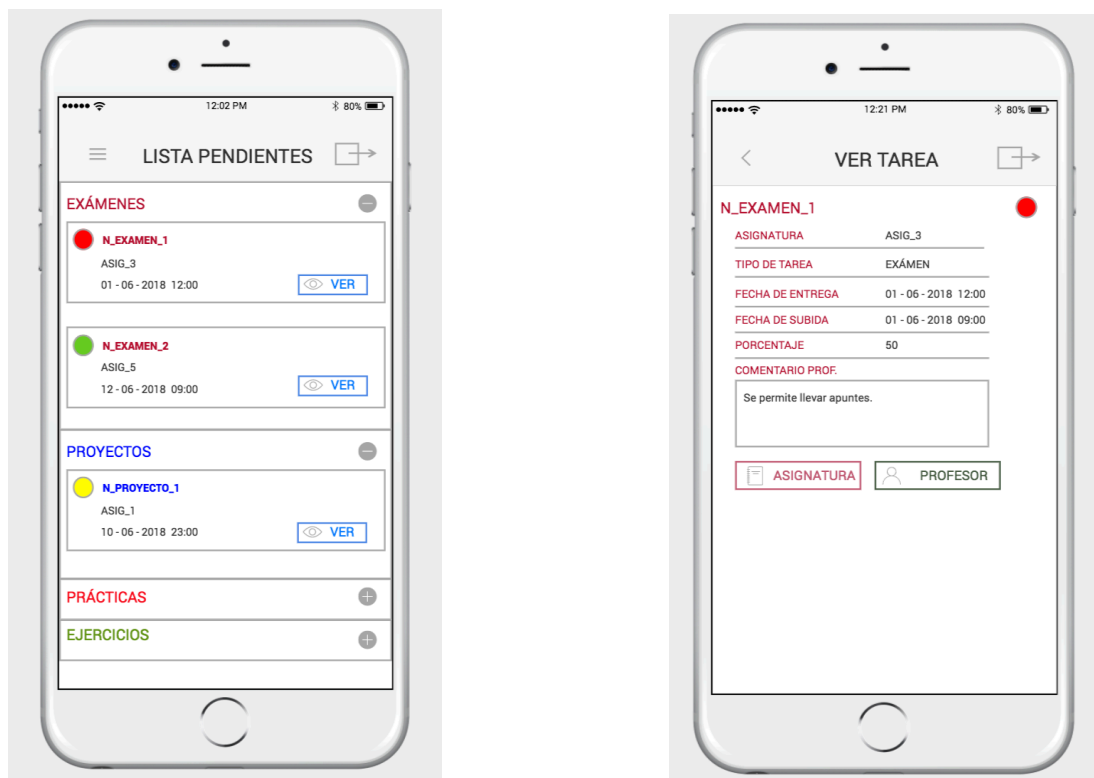


Figura 14: Boceto APP - Lista de elementos y ver elemento

Las heurísticas “*Flexibilidad y eficiencia de uso*” y “*Diseño estético y minimalista*” están conectadas. Si el diseño de una aplicación no está recargado de información innecesaria, el usuario puede conocer mejor la funcionalidad de la aplicación. Por este motivo, las aplicaciones desarrolladas solo muestran la información necesaria para cada opción, en forma de tabla para las aplicaciones web y en forma de lista para la aplicación para dispositivos móviles. Ambas usan botones para moverse por la aplicación.

6. FUNCIONALIDAD DE LA APLICACIÓN

En esta sección se detallarán los pasos que hay que seguir para navegar en la aplicación y realizar las tareas en función del tipo de usuario. Para comprender mejor el funcionamiento, este apartado se dividirá en pantallas y se mostrará una captura de cada situación para completar el manual.

Para acceder al código desarrollado durante el proyecto, se introduce <https://github.com/adriver91/tfgOrganizaMisPracticas> en el se encuentra el archivo Readme.txt con las instrucciones necesarias para ejecutar cada aplicación.

6.1 Manual del administrador

En este apartado se detallarán los pasos que debe realizar el administrador para realizar cambios sobre la base de datos, salvo las tablas relacionadas con las tareas de los alumnos, que solamente podrán ser administradas por el profesorado. Dichos cambios podrán ser: ver una lista de datos, ver un dato concreto, añadir una nueva fila a una tabla existente, y editar o eliminar un dato específico.

6.1.1 Login

El administrador la dirección correspondiente en el navegador, al abrirla se muestra una primera pantalla, se puede observar en la Figura 15, con la opción de loguearse. El administrador se loguea introduciendo el correo y la contraseña correctos. En caso de que se produzca un error, se muestra un mensaje informativo.

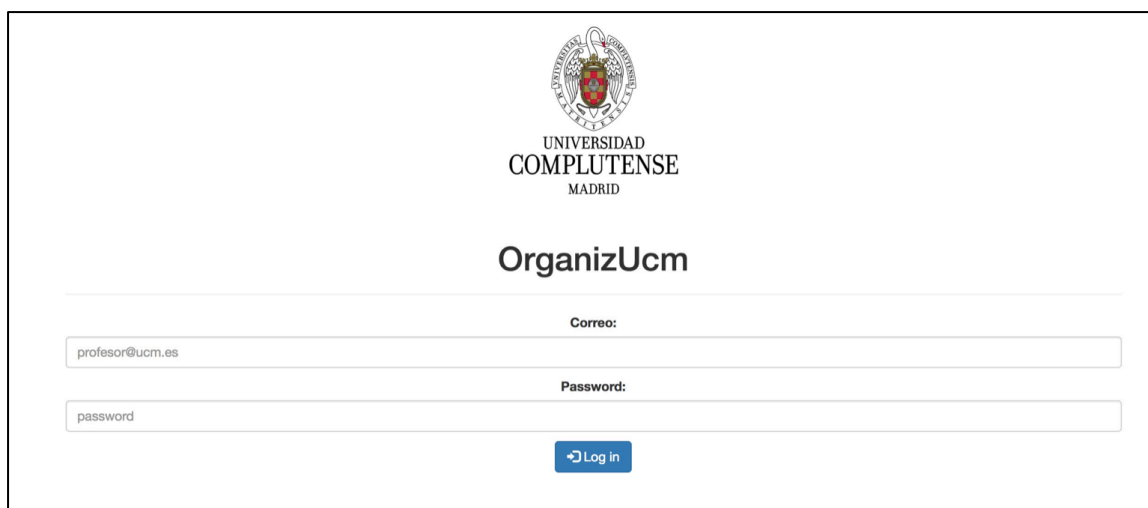
The image shows a web login interface. At the top center is the coat of arms of the Universidad Complutense de Madrid, with the text 'UNIVERSIDAD COMPLUTENSE MADRID' below it. Underneath is the title 'OrganizUcm'. There are two input fields: the first is labeled 'Correo:' and contains the text 'profesor@ucm.es'; the second is labeled 'Password:' and contains the text 'password'. Below these fields is a blue button with a white arrow icon and the text 'Log in'.

Figura 15: Log in - web

6.1.2 Página principal

Después de loguearse, el administrador será dirigido a la página principal, que se puede observar en la Figura 16. En ella se muestran las tablas sobre las cuales el administrador puede actuar.



Figura 16: Página principal administrador web

En la cabecera de la página se pueden apreciar dos botones situados a ambos lados del título, se pueden observar en la Figura 17. En el lado derecho aparece el botón que representa la salida de la aplicación. En el lado izquierdo aparece el botón que representa un menú.



Figura 17: Botón menú y botón salir

El botón para salir de la aplicación redirige al administrador a la página de login que se ha mostrado en la Figura 15. Este botón se muestra en las páginas donde se representa una lista de datos o la información de un dato concreto.

El botón de menú muestra una barra lateral. La opción “X” oculta la barra lateral. La opción “*Editar contraseña*” redirigirá al administrador a una nueva página que se explica en el apartado 6.1.3.

La opción “*Acceder*” es un submenú desplegable con tres opciones que dirigirán al administrador a otras páginas donde se muestran las relaciones entre las tablas sobre las que el administrador puede actuar. De manera que en cada una de ellas se pueda ver la lista de los elementos de la relación o la información de una relación concreta, o añadir, editar o eliminar una relación. Dichas relaciones son “*Asignaturas – Alumnos*”, “*Profesores – Asignaturas*” y “*Profesores - Facultades*” .

La opción “*Otras relaciones*” es un submenú desplegable con dos opciones que dirigirán al administrador a dos páginas, una por cada opción, donde el administrador podrá ver la lista de los elementos de cada relación que representa cada opción. “*Facultades - Alumnos*” dirigirá al administrador a una página que muestra la lista de alumnos pertenecientes a las distintas facultades y al tipo de estudio al que pertenecen, ordenadas por nombre de facultad. El segundo “*Profesores – Alumnos*” se comporta de la misma forma que el otro botón, pero mostrando la lista de alumnos con sus respectivos profesores y la asignatura que estos últimos les imparten, ordenado por el correo del profesor.

6.1.3 Editar contraseña

Desde el menú de la pantalla principal se accede a la página, se puede observar en la Figura 18, que permite al administrador editar su contraseña. Para poder realizar este proceso, inicialmente el administrador deberá introducir su antigua contraseña y a continuación escribir su nueva contraseña y confirmarla. En caso de error, se muestra un mensaje informativo.



Figura 18: Página "Editar contraseña" - web

En la cabecera de la página se encuentran dos botones. El botón del lado izquierdo, anteriormente mostrado en la Figura 17, representa la salida de la aplicación. El botón de la derecha, que se puede observar en la Figura 19, representa el “Home” y dirige al administrador a la página principal. Estos dos botones aparecerán en todas las páginas referentes al administrador, cuya finalidad sea mostrar una lista de datos.



Figura 19: Botón home

6.1.4 Mostrar lista de datos

Desde la ventana principal, se puede acceder las tablas sobre las cuáles el administrador puede actuar. Cuando el administrador pulsa el nombre de una de las tablas, se dirige a una nueva página donde se muestra una lista con la información contenida en dicha tabla, como se puede observar en Figura 20.

En la cabecera de la página se observan dos botones anteriormente mencionados. A la izquierda se encuentra el botón “Home”, que se ha mostrado en la Figura 19. A la derecha se encuentra el botón “Salir”, se ha mencionado en la Figura 17. Ambos cumplen su funcionalidad anteriormente mencionada.

En la página se observa una lista de datos con la información más relevante sobre la tabla de datos y un botón en la esquina superior izquierda de la tabla, se utiliza para crear un nuevo elemento.

Ver asignaturas

Crear asignatura

Abreviatura	Nombre	Creditos	Curso	Grupo	Cuatrimestre	Estudios	Acciones		
a_9	n_asign9	6	3	A	segundo	G_f1_1	Ver	Editar	Borrar
a_8	n_asign8	6	3	A	segundo	G_f1_1	Ver	Editar	Borrar
a_7	n_asign7	6	3	A	segundo	G_f2_1	Ver	Editar	Borrar
a_6	n_asign6	6	2	A	primer	G_f2_1	Ver	Editar	Borrar
a_5	n_asign5	9	1	A	anual	G_f1_2	Ver	Editar	Borrar
a_4	n_asign4	12	2	A	anual	G_f1_1	Ver	Editar	Borrar
a_3	n_asign3	6	3	A	segundo	G_f1_1	Ver	Editar	Borrar
a_2	n_asign2	6	4	A	primer	G_f1_1	Ver	Editar	Borrar
a_1	n_asign1	6	1	A	primer	G_f1_1	Ver	Editar	Borrar

Figura 20: Página "Lista de elementos" - web



Figura 21: Acciones sobre un elemento

Al final de la página se observa una barra de paginación numérica para ver la información completa del contenido de la tabla. Por defecto, el número máximo de filas por tabla es de 5. La barra muestra con color oscuro la posición en la que se encuentra el administrador, con un tono más claro las posiciones restantes y con un color blanco la primera y la última posición, facilitando la movilidad al administrador.

6.1.5 Crear un nuevo dato

Desde la ventana “*Lista de elementos*” se accede a la página “*Crear elemento*”, que se muestra en la Figura 22, cuando se pulsa el botón de “*Crear elemento*”, mencionado en el apartado 6.1.4.

En la página se observa el botón “*Ver elementos*”, en la imagen se muestra como “*Ver asignaturas*”, en la esquina superior derecha del formulario. Dirige al administrador a la lista de elementos mencionado en el apartado 6.1.4, que se observa en la Figura 20.

Se muestra un formulario que se compone de las columnas de la tabla sobre la que actúa el administrador. El administrador debe introducir los valores con el tipo de datos correcto, en caso de no hacerlo correctamente cuando pulse el botón “*Crear*” se muestra un mensaje como el de la Figura 23. Además, el administrador debe cumplir las restricciones de cada tipo, si no cuando se pulse el botón “*Crear*”, se muestra un mensaje como el de la Figura 24, bajo el dato donde se ha producido el fallo.

Crear asignatura

[Ver asignaturas](#)

Abreviatura	
Nombre	
Creditos	
Curso	
Grupo	
Cuatrimestre	
Estudios	<div style="border: 1px solid #ccc; padding: 2px;"> Seleccionar estudio... </div>

[Crear](#)

Figura 22: Página "Crear elemento" - web

No se pudo crear una asignatura.

Figura 23: Error sobre tipo de datos - web

!

El valor debe ser inferior o igual a 12

Figura 24: Error restrictivo - web

Algunos datos solo se pueden insertar si hacen referencia a otros datos existentes, de manera que el administrador selecciona la opción. Por ejemplo, en la Figura 22 se hace referencia a la creación de una nueva asignatura. Una asignatura nueva solo se puede crear en relación a un tipo de estudio existente, como puede ser “*Grado en Ingeniería Informática*”.

Cuando el administrador termina de introducir los datos, pulsa el botón “*Crear*”. Si los datos son correctos se muestra un mensaje de confirmación, en caso de error, se muestra un mensaje, que se observa en la Figura 23, indicando que se ha producido un error en el proceso.

6.1.6 Ver información sobre un dato

Desde la ventana “*Lista de elementos*” se observa que por cada fila hay una serie de acciones asociadas a ella. Cuando se pulsa el botón de “*Ver*” de una fila determinada, se accede a la página “*Ver un elemento*”, que se muestra en la Figura 25. Donde se observa su información detallada.

Ver una asignatura	
Abreviatura	a_6
Nombre	n_asign6
Creditos	6
Curso	2
Grupo	A
Estudios	G_f2_1

Figura 25: Página "Ver un elemento" - web

En la cabecera de la página se muestran los botones para ir a la página principal y salir de la aplicación, mencionados anteriormente. En la página se puede observar el botón “*Ver elementos*”, en la imagen se muestra como “*Ver asignaturas*”, en la esquina superior derecha de la tabla, dirige al administrador a la lista de elementos mencionada en el apartado 6.1.4, que se observa en la Figura 20.

Debajo del botón, se muestra una tabla con los datos sobre el elemento seleccionado. Desde dicha página se puede acceder a la información sobre el elemento con que se relaciona, por ejemplo, en la Figura 25, se muestra la información de una asignatura y desde dicha tabla se accede a la información del tipo de estudio al que pertenece. Dicho comportamiento se produce siempre que los elementos dependan de otras tablas.

6.1.7 Editar información de un dato

Desde la ventana “*Lista de elementos*” se observa que por cada fila hay una serie de acciones asociadas a ella. Cuando se pulsa el botón de “*Editar*” de una fila determinada, se accede a la página “*Editar un elemento*”, que se muestra en la Figura 26.

Editar asignatura	
Abreviatura	<input type="text" value="a_6"/>
Nombre	<input type="text" value="n_asign6"/>
Creditos	<input type="text" value="6"/>
Curso	<input type="text" value="2"/>
Grupo	<input type="text" value="A"/>
Cuatrimestre	<input type="text" value="primer"/>
Estudios	<input type="text" value="G_f2_1"/>
<input type="button" value="Editar"/>	

Figura 26: Página "Editar elemento" - web

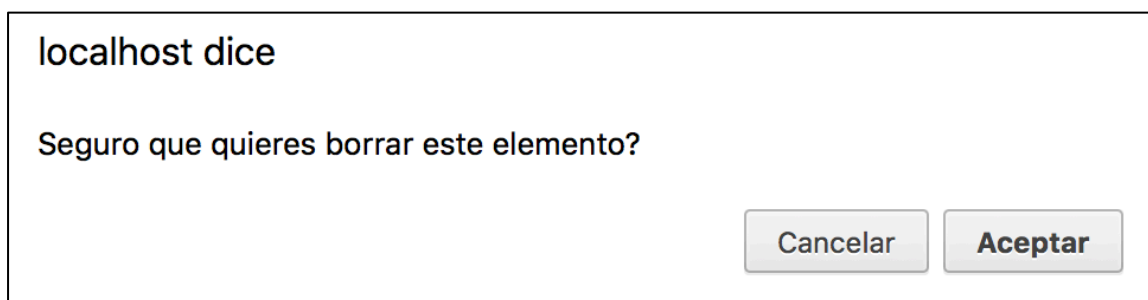
En la página se puede observar el botón “*Ver elementos*”, en la imagen se muestra como “*Ver asignaturas*”, en la esquina superior derecha del formulario. Dirige al administrador a la lista de elementos mencionado en el apartado 6.1.4, que se observa en la Figura 20.

Debajo del botón, se muestra un formulario autocompletado con la información del elemento a editar, de manera que el administrador puede ver la información detallada del elemento y modificar la información que él considere necesaria. Para poder realizar los cambios, el administrador debe pulsar el botón “*Editar*” que se encuentra situado al final del formulario.

Al igual que para crear un nuevo elemento, la información que introduzca el administrador debe ser correcta en cuanto a tipos de datos y restricciones. Cuando el administrador termina de modificar los datos, pulsa el botón “*Editar*”. Si los datos son correctos se muestra un mensaje de confirmación. En caso de error, se muestra un mensaje similar al que se muestra en la Figura 23, indicando que se ha producido un error en el proceso o que el administrador ha introducido mal los datos.

6.1.8 Borrar un dato

Desde la ventana “*Lista de elementos*”, cuando el administrador pulsa el botón de “*Borrar*” aparece el mensaje que se muestra en la Figura 27. Dicho mensaje se muestra por si el administrador comete un error al pulsar el botón. Si el administrador cancela dicho mensaje, se mantiene en la página “*Lista de elementos*”.



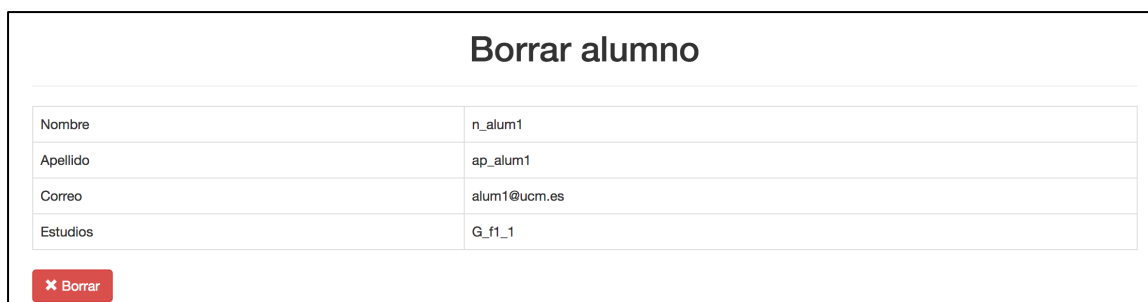
localhost dice

Seguro que quieres borrar este elemento?

Cancelar Aceptar

Figura 27: Mensaje confirmación para borrar un elemento

Si el administrador acepta el mensaje, se dirige a la página “*Borrar elemento*”, se muestra en la Figura 28. En dicha página se muestra la información del elemento a eliminar y se elimina el elemento mostrado. Cuando el administrador pulsa sobre el botón “*Borrar*”, se dirige a la ventana “*Lista de elementos*”.



Borrar alumno

Nombre	n_alum1
Apellido	ap_alum1
Correo	alum1@ucm.es
Estudios	G_f1_1

Figura 28: Página "Borrar un elemento" - web

Cabe destacar que las relaciones que aparecen en la opción “*Acceder*” del menú, mencionadas en el apartado 6.1.2, poseen las mismas acciones anteriormente mencionadas. En la relación “*Asignaturas - Alumnos*”, el administrador puede ver la lista de alumnos y sus respectivas asignaturas, ver la información de matriculación de un alumno respecto a una asignatura, añadir un nuevo alumno a una asignatura, editar la información de matriculación y eliminar a un alumno de una asignatura. El resto de relaciones tienen un comportamiento similar en función de su relación.

Mientras que las relaciones que aparecen en la opción “*Otras relaciones*”, mencionadas en el apartado 6.1.2, solo muestra la lista con la información de los elementos de la relación. Estas relaciones solo son de carácter informativo.

6.2 Manual del profesorado

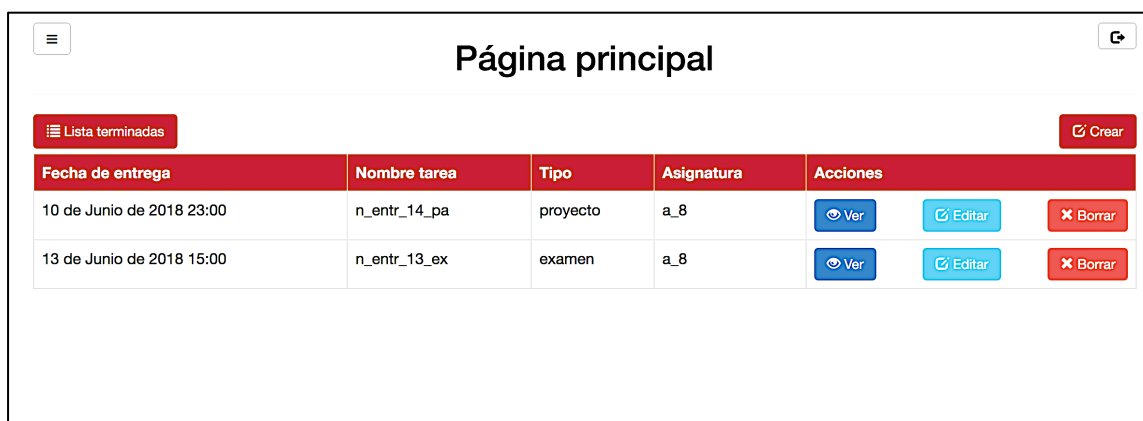
En este apartado se detallan los pasos que debe realizar un profesor para realizar cambios sobre la base de datos. Dichos cambios se efectuarán sobre las tareas de las asignaturas impartidas por dicho profesor, incluyendo la evaluación de esas tareas. Se divide por pantallas y se muestra una captura de cada situación para facilitar la guía. La visualización de las acciones es muy similar a la mostrada en la web del administrador, pues en ambas, como se ha mencionado en el apartado 5.1, se utiliza el mismo diseño de interfaz.

6.2.1 Login

El profesor introduce la dirección web correspondiente en el navegador. Al abrirla se muestra una primera pantalla, se puede observar en la Figura 15, con la opción de loguearse. El profesor se loguea introduciendo el correo y la contraseña correctos. En caso de que se produzca un error, se muestra un mensaje informativo.

6.2.2 Página principal

Después de loguearse, el profesor será dirigido a la página principal, que se puede observar en la Figura 29. En la cabecera de la página se pueden apreciar dos botones situados a ambos lados del título, se pueden observar en la Figura 17. En el lado derecho aparece el botón que representa la salida de la aplicación. En el lado izquierdo aparece el botón que representa un menú.



Página principal				
Lista terminadas				Crear
Fecha de entrega	Nombre tarea	Tipo	Asignatura	Acciones
10 de Junio de 2018 23:00	n_entr_14_pa	proyecto	a_8	Ver Editar Borrar
13 de Junio de 2018 15:00	n_entr_13_ex	examen	a_8	Ver Editar Borrar

Figura 29: Página principal web del profesor - web

El botón de menú muestra una barra lateral. La opción “X” oculta la barra lateral. La opción “*Editar contraseña*” dirige al profesor a una nueva página que se ha explicado en el apartado 6.1.3.

La opción “*Lista asignaturas*” dirige al profesor a una nueva página donde se muestra la lista de las asignaturas que imparte el profesor. La opción “*Lista alumnos*” dirige al profesor a una nueva página donde se muestra la lista de los alumnos que se encuentran en las clases que imparte el profesor. La opción “*Información personal*” dirige al profesor a una nueva página donde se muestra sus datos personales.

En la página se observa una tabla de datos con la información más relevante sobre las tareas que aún no han finalizado, ordenadas por asignatura, y encima de dicha tabla se encuentran dos botones. En la esquina superior izquierda se encuentra el botón “*Lista terminadas*” que dirige al profesor a la lista de tareas que han finalizado. En la esquina superior derecha, se encuentra el botón “*Crear*” que se utiliza para crear una nueva tarea.

Cada fila posee tres botones, los mismos que se pueden ver en la Figura 21, se representan las posibles acciones a realizar en función de la tarea referenciada que se muestra en la fila. El botón “*Ver*”, dirige al profesor a una nueva página donde se ve la información sobre una tarea seleccionada. El botón “*Editar*”, dirige al profesor a una nueva página donde se modifica la información sobre una tarea seleccionada. Y el botón “*Borrar*”, dirige al profesor a una nueva página donde se borra la entrega.

6.2.3 Crea tarea

Desde la venta principal se accede a la página “*Crear tarea*”, cuando se pulsa el botón “*Crear*”, mencionado en el apartado 6.2.2. La página posee el mismo comportamiento que el apartado 6.1.5.

Se muestra un formulario que se compone de la información necesaria que tiene que introducir un profesor para crear una nueva tarea. En la página se observa el botón “*Ver tareas*”, en la esquina superior derecha del formulario. El botón dirige al profesor a la página principal donde se encuentran las tareas pendientes. En el formulario, el profesor debe introducir los valores con el tipo de datos correcto, en caso de no hacerlo correctamente cuando pulse el botón “*Crear*” se muestra un mensaje como el de la Figura 23. Además, el profesor debe cumplir las restricciones de cada tipo, si no cuando se pulse el botón “*Crear*”, se muestra un mensaje como el de la Figura 24, bajo el dato donde se ha producido el fallo.

6.2.4 Editar contraseña

Posee la misma funcionalidad y representación que la que se encuentra en el apartado 6.1.3.

6.2.5 Lista de tareas terminadas

Desde la ventana principal, se accede a la página “*Lista de tareas terminadas*”, que se muestra en la Figura 30, cuando se pulsa el botón “*Lista terminadas*”, mencionado en el apartado 6.2.2.

En la cabecera de la página se observan dos botones. A la izquierda se encuentra el botón “*Home*”, que es el mismo que se ha utilizado en la Figura 19, el cuál dirige al profesor a la página principal. A la derecha se encuentra el botón “*Salir*”, dirige al profesor a la pantalla de login.

En la página se observa una lista de datos, representada en una tabla ordenadas por nombre de asignatura. Las columnas muestran el nombre de la tarea, el tipo al que pertenece, la asignatura y la nota media de los alumnos en dicha tarea.

Lista tareas terminadas						
Fecha de entrega	Nombre tarea	Tipo	Asignatura	Nota	Acciones	
30 de Enero de 2018 11:00	n_entr_1_ex	examen	a_1	52.5000	 Ver	 Evaluar
15 de Noviembre de 2017 23:00	n_entr_2_pa	practica	a_1	50.0000	 Ver	 Evaluar
1 de Junio de 2018 12:00	n_entr_7_ex	examen	a_4	Sin evaluar	 Ver	 Evaluar
7 de Febrero de 2018 15:00	n_entr_6_ex	examen	a_4	57.5000	 Ver	 Evaluar
1 de Junio de 2018 8:00	n_entr_8_po	proyecto	a_5	Sin evaluar	 Ver	 Evaluar
4 de Febrero de 2018 12:00	n_entr_9_ex	examen	a_5	37.0000	 Ver	 Evaluar
5 de Febrero de 2018 17:00	n_entr_10_ex	examen	a_6	60.0000	 Ver	 Evaluar
5 de Enero de 2018 23:00	n_entr_11_pa	practica	a_6	50.0000	 Ver	 Evaluar

Figura 30: Página "Lista tareas terminadas" - web

Cada fila posee dos botones. El botón “*Evaluar*”, que se puede ver en la Figura 31, dirige al profesor a una nueva página donde evalúa la tarea que pertenece a la fila donde se pulsó el botón. El botón “*Ver*”, mostrado en la Figura 21, dirige al profesor a una nueva página donde se muestra la información sobre la tarea seleccionada y la media de los alumnos en dicha tarea.



Figura 31: Botón evaluar

6.2.6 Evaluar tarea

Desde la ventana “*Lista de tareas terminadas*”, se accede a la página “*Evaluar tarea*”, que se muestra en la Figura 32, cuando se pulsa el botón “*Evaluar*”, mencionado en el apartado 6.2.6, perteneciente a una tarea que se sitúa en una fila de la tabla.

Evaluar tarea				
n_entr_2_pa [a_1]				
Nombre	Nota	Comentario	Acciones	
n_alum1 ap_alum1	85		 Ver	 Editar
n_alum2 ap_alum2	15	Revision: proximo lunes a las 10 en mi despacho	 Ver	 Editar

Figura 32: Página "Evaluar tareas" - web

En la cabecera de la página se observan dos botones. A la izquierda se encuentra el botón “*Home*”, dirige al profesor a la página principal. A la derecha se encuentra el botón “*Salir*”, dirige al profesor a la pantalla de login.

En la página se observa una tabla de datos con el nombre y apellidos de los alumnos que están matriculados en la asignatura a la que dicha tarea pertenece, la nota y el comentario sobre dicha calificación. La tabla se encuentra ordenada en orden alfabético en relación al apellido de los estudiantes. En la esquina superior izquierda de la tabla, se muestra el nombre de la tarea que se evalúa y la asignatura a la que pertenece dicha tarea.

Cada fila posee tres botones, mostrados en la Figura 21. El botón “*Ver*” dirige al profesor a una nueva página donde se muestra la información sobre la tarea y el alumno referente a la fila seleccionada. El botón “*Editar*” comprueba si el valor de la columna “*Nota*” está vacío. Si el valor es vacío, introduce la calificación, en caso contrario, edita la calificación existente. La columna “*Comentario calificación*” posee el mismo comportamiento cuando se pulsa “*Editar*”. El botón “*Borrar*” dirige al profesor a una nueva página donde se elimina la calificación para dicho estudiante en esa tarea.

6.2.7 Ver lista de asignaturas y ver lista de alumnos

Desde la opción del menú de la página principal, se accede a la página “*Lista de asignaturas*” y a “*Lista de alumnos*”, cuando se pulsa la opción “*Lista asignaturas*” y “*Lista alumnos*”, respectivamente, mencionados en el apartado 6.2.2.

En la cabecera de las páginas se muestran dos botones. A la izquierda se encuentra el botón “*Home*”, que es el mismo que se ha utilizado en la Figura 19, el cuál dirige al profesor a la página principal. A la derecha se encuentra el botón “*Salir*”, que es el mismo que se ha utilizado en la Figura 17, dirige al profesor a la pantalla de login.

En la página “*Lista de asignaturas*” se muestra una tabla con las asignaturas que el profesor imparte y la información más relevante a cada una de ellas. Contiene el número de alumnos matriculados en cada una de ellas y la media de ellos. La tabla posee un botón “*Ver*” por cada fila, el botón dirige al profesor a una nueva página donde se muestra la información detallada de la asignatura.

En la página “*Lista de alumnos*” se muestra una tabla con la información de los alumnos pertenecientes a las asignaturas que el profesor imparte, ordenada por asignatura. Contiene la calificación hasta el momento dicho alumno en la asignatura. La tabla posee un botón “*Ver*” por cada fila, el botón dirige al profesor a una nueva página donde se muestra la información detallada del alumno

6.2.8 Ver información de un dato, editar información de un dato y eliminar un dato

Cada opción dirige al profesor a una nueva página en función de la acción a realizar y del lugar desde que se ha realizado dicha llamada. A continuación se detalla el comportamiento de las acciones dependientes del lugar.

La acción “*Ver información*” de un dato muestra una tabla con los contenidos relevantes sobre dicho dato. La visualización de las páginas es igual a la mostrada en la Figura 25. Además, incluye los botones de la cabecera mencionados en la Figura 19 que hace referencia a “*Home*”, y la Figura 17 que hace referencia a “*Salir*”. Si el profesor se encuentra en la página principal, la página a la que se dirige muestra la información sobre la tarea no finalizada seleccionada en la fila. Si se encuentra en la página “*Lista terminadas*”, la página a la que se dirige muestra la información sobre la tarea finalizada seleccionada en la fila, incluyendo la nota media de los alumnos en dicha tarea. Si se sitúa en la página “*Evaluar tarea*”, la nueva página muestra la información de la tarea y del alumno pertenecientes a la fila seleccionada. Si se sitúa en “*Lista asignaturas*”, dirige al profesor a una nueva página donde se muestra la información detallada de la asignatura, incluyendo el número de alumnos matriculados y una tabla con la lista de las tareas pertenecientes a dicha asignatura y la nota media de ellas. Si se sitúa en “*Lista alumnos*”, dirige al profesor a una nueva página donde se muestra la información detallada del alumno seleccionado en la fila y la lista de las tareas que tiene el alumno en las que el profesor le da clase y la nota en cada una de ellas.

Si el profesor se encuentra en la página principal y pulsa el botón “*Editar*”, la página a la que se dirige muestra un formulario autocompletado con los contenidos de la tarea no finalizada seleccionada en la fila que se desea editar. Se muestra un botón en la esquina superior derecha del formulario que redirige al profesor a la página principal. La visualización y el comportamiento de dicha página es la misma a la mostrada en la Figura 27, mencionado en el apartado 6.1.7. Si se sitúa en la página “*Evaluar tarea*”, el funcionamiento se encuentra en el apartado 6.2.6.

La acción “*Eliminar un dato*” muestra una tabla con el contenido del dato a eliminar. En todos los casos, se muestra un mensaje de confirmación antes de realizar la acción, como el que aparece en la Figura 27. Si cancela dicho mensaje, el profesor se mantendrá en la página donde se pulso el botón “*Borrar*”. Si acepta dicho mensaje, el profesor se dirige a una nueva página. Si se encuentra en la página principal, se dirige a una nueva página donde se borra la entrega no finalizada seleccionada en la fila. Si se sitúa en la página “*Evaluar tarea*”, se dirige a una nueva página donde se elimina la calificación del estudiante seleccionado en la fila sobre la tarea seleccionada. En ambos casos, cuando pulse el botón “*Borrar*” el profesor se dirige a la página de donde procedía. La visualización y comportamiento es el mismo al mostrado en la Figura 28.

6.2.9 Información personal

Desde la opción del menú de la página principal, se accede a la página “*Información personal*”, cuando se pulsa la opción “*Información personal*”, mencionado en el apartado 6.2.2.

En la cabecera de las páginas se muestran dos botones. A la izquierda se encuentra el botón “*Home*”, que es el mismo que se ha utilizado en la Figura 19, el cuál dirige al profesor a la página principal. A la derecha se encuentra el botón “*Salir*”, que es el mismo que se ha utilizado en la Figura 17, dirige al profesor a la pantalla de login.

En la página se muestra en una tabla, la información del profesor, incluyendo la información de las facultades donde trabaja. En la esquina superior derecha de la tabla se encuentra un botón “*Editar contraseña*” que dirige al profesor a la página donde puede modificar su contraseña.

6.3 Manual del estudiante

En este apartado se detallan los pasos que debe realizar un alumno para navegar por la aplicación para dispositivos móviles. Se divide por pantallas y se muestra una captura de cada situación para facilitar la guía.

6.3.1 Login

El alumno abre la aplicación en su teléfono, al abrirla se muestra una primera pantalla, se observa en la Figura 33, con la opción de loguearse. El alumno se loguea introduciendo su correo de la universidad y la contraseña correcta. En caso de que se produzca un error, se muestra un mensaje informativo.

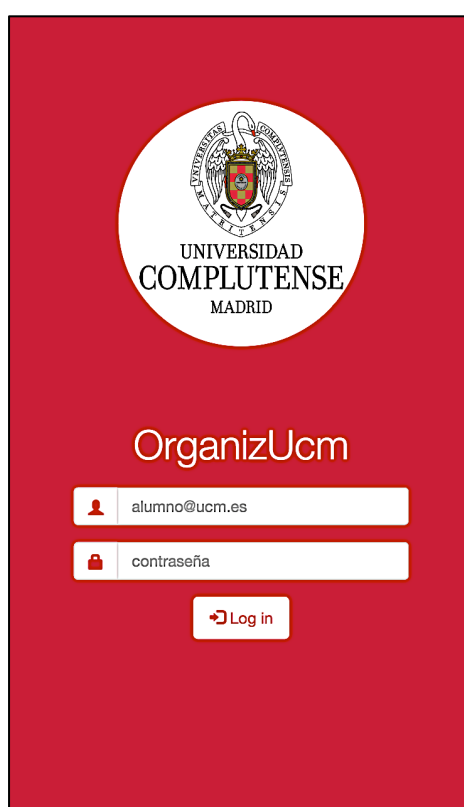


Figura 33: Log in - móvil

6.3.2 Página principal

Después de loguearse, el profesor será dirigido a la página principal, que se puede observar en la Figura 34. En la cabecera de la página se pueden apreciar dos botones situados a ambos lados del título, ambos se pueden observar en la Figura 17. En el lado derecho aparece el botón que representa la salida de la aplicación, que dirige al alumno a la página de Login mostrada en la figura 33. En el lado izquierdo aparece el botón que representa un menú, se observa en la Figura 35.

El botón de menú muestra una barra lateral. La opción “X” oculta la barra lateral. La opción “*Editar contraseña*” redirigirá al estudiante a una nueva página que edita la contraseña que utiliza el alumno en la aplicación. La opción “*Asignaturas*” dirige al alumno a la página “*Lista de asignaturas*”, donde se muestran las asignaturas que el

estudiante está matriculado. La opción “*Profesores*” dirige al alumno a la página “*Lista de profesores*”, donde se muestran los que imparten clase al alumno. La opción “*Datos personales*” dirige al alumno a una página donde se muestra la información personal del alumno.

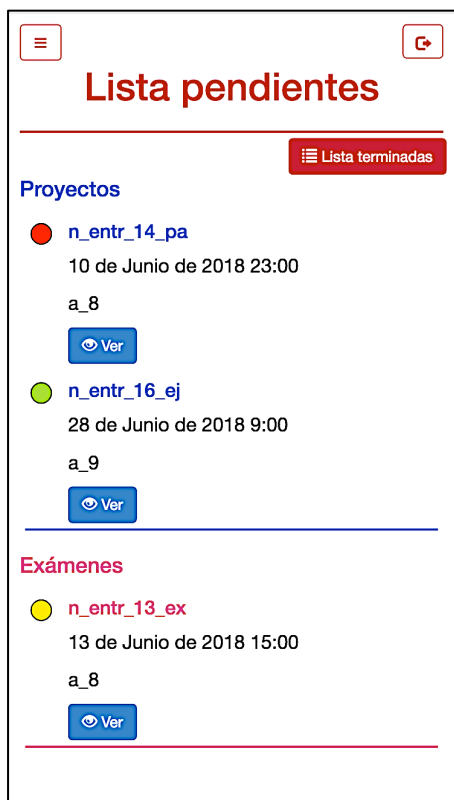


Figura 34: Página principal - móvil

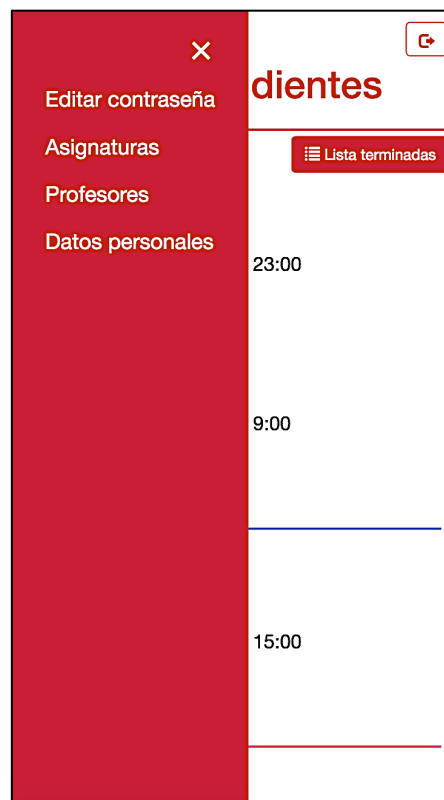


Figura 35: Menú - móvil

En la página se observa una lista de datos y un botón en la esquina superior izquierda de la tabla. El botón “Lista terminadas” dirige al alumno a una nueva página donde se muestra la lista de las tareas que han finalizado.

La lista que se muestra en la página esta organizada en grupos por el tipo de tarea. Cada grupo esta señalado con el nombre correspondiente al tipo al que pertenece, es decir, están ordenados por “*Proyectos*”, “*Exámenes*”, “*Prácticas*” y “*Ejercicios*”, cada uno de ellos representado por un color, para facilitar la lectura. La tarea cuya fecha de entrega sea la más cercana, se muestra en primera posición, y por tanto, todo el grupo al que pertenece. Cada grupo esta dividido en subgrupos, o tareas, ordenados por fecha límite de entrega.

Dentro de cada subgrupo se muestra el nombre de la tarea con el color del grupo al que pertenece, la fecha límite de entrega y la asignatura a la que pertenece dicha entrega. Al lado del nombre se encuentra un círculo, representa el plazo a entregar dicha tarea. Si el círculo es de color verde, el alumno posee más de una semana para entregar la tarea; si es amarillo tendrá una semana o menos y si es rojo, tendrá menos de tres días. Además, en cada subgrupo se encuentra un botón “*Ver*”, mostrado en la Figura 21, que dirige al alumno a una nueva página donde se observa la información detallada de la tarea.

6.3.3 Ver información de una tarea pendiente

Desde la ventana principal se accede a la página “*Ver tarea pendiente*”, que se muestra en la Figura 36, cuando se pulsa el botón de “*Ver*” situado en un subgrupo.

Nombre	n_entr_13_ex
Fecha de inicio	13 de Junio de 2018 12:00
Fecha limite	13 de Junio de 2018 15:00
Tipo de entrega	examen
Porcentaje	40
Asignatura	a_8

Asignatura Profesor

Figura 36: Ver tarea pendiente - móvil

En la cabecera de la página se observan dos botones anteriormente mencionados. A la izquierda se encuentra el botón “*Home*”, que se ha mostrado en la Figura 19, que dirige al alumno a la página principal. A la derecha se encuentra el botón “*Salir*”, se ha mencionado en la Figura 17, que dirige al alumno a la pantalla del Login.

En la página se muestra una tabla con la información de la tarea pendiente y el círculo que le corresponde. Al final de la lista se encuentran dos botones, “*Asignatura*” y “*Profesor*”. Si el alumno pulsa el primer botón, se dirige a la página “*Ver asignatura*” que muestra la información de la asignatura a la que pertenece dicha tarea. Si el alumno pulsa el segundo botón, se dirige a la página “*Ver profesor*” que muestra la información del profesor que imparte la asignatura a la que pertenece dicha entrega.

6.3.4 Ver lista de tareas terminadas

Desde la ventana principal se accede a la página “*Lista terminadas*” cuando se pulsa el botón “*Terminadas*”. En la cabecera se encuentran el botón “*Home*” y “*Salir*”, anteriormente mencionados. La página muestra una estructura similar a la página principal. Las tareas también se dividen en grupos y subgrupos. Las únicas diferencias relevantes se encuentran en la información del grupo.

En un subgrupo se muestra el nombre de la tarea, marcada con el color de su grupo, la asignatura a la que pertenece y la calificación. Al lado del nombre de la tarea se encuentra un cuadrado. Si el cuadrado es de color azul claro, aún no ha sido calificada dicha tarea. Si es de color verde, significa que el alumno ha aprobado la entrega y si es rojo, la ha suspendido. También cuenta con un botón “Ver”, mostrado en la Figura 21, que dirige al alumno a una nueva página donde se muestra la información de la tarea terminada.

6.3.5 Ver información de una tarea terminada

Desde la ventana “Lista terminadas” se accede a la página “Ver tarea terminada”, se observa en la Figura 37, cuando se pulsa el botón de “Ver” situado en un subgrupo. La Figura 37 se compone de dos imágenes, la primera con la evaluación aprobada y el comentario del profesor. La segunda sin comentario ni calificación. En la cabecera se encuentran el botón “Home” y “Salir”, anteriormente mencionados.

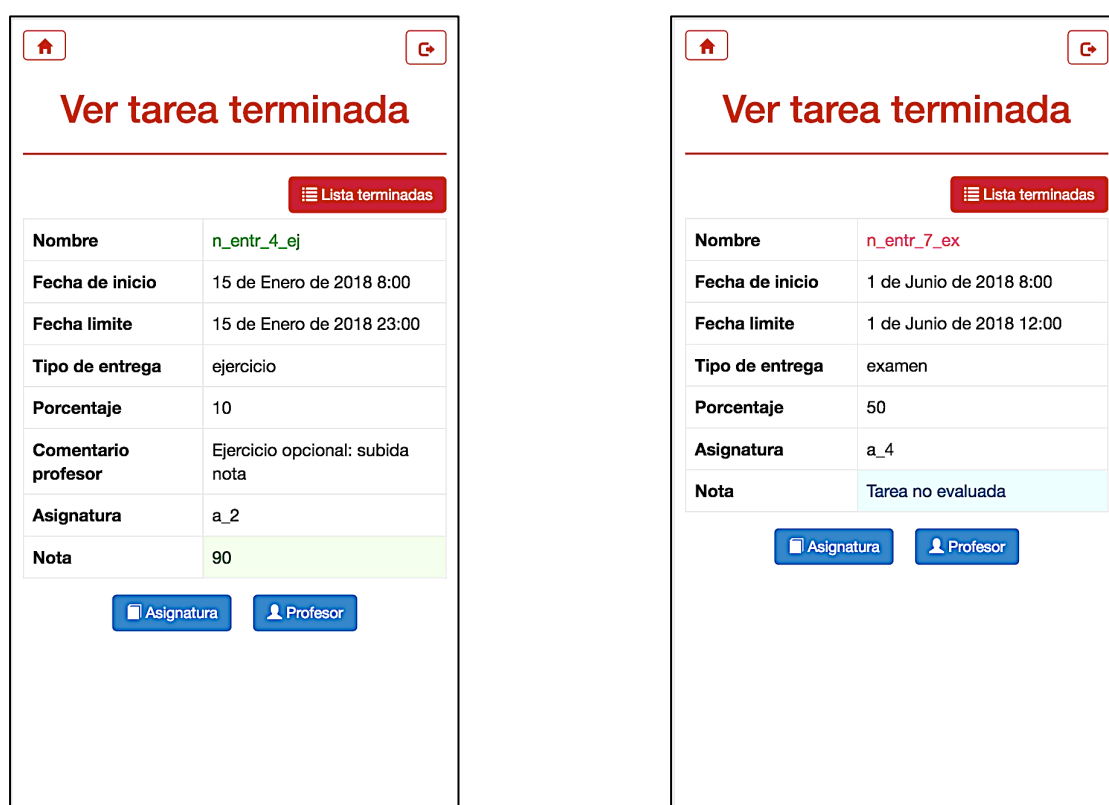


Figura 37: Tarea terminada - móvil

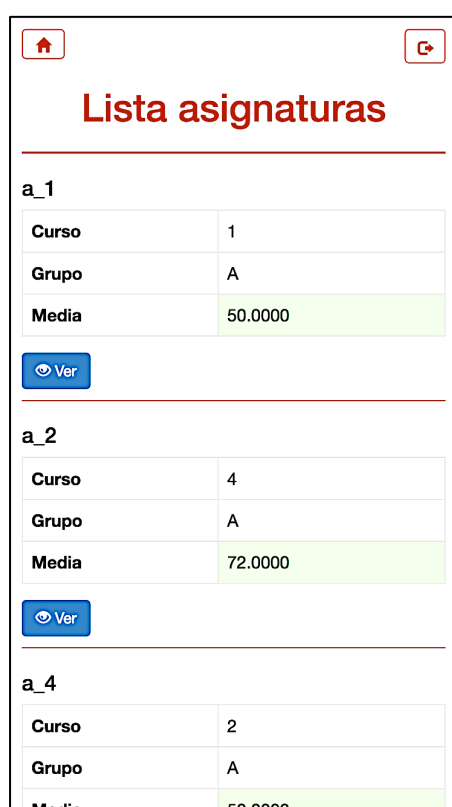
En la esquina superior izquierda de la tabla se muestra el botón “Lista terminadas”, que dirige al alumno a una nueva página donde se muestra la lista de las tareas que han finalizado.

En la página se muestra una tabla con la información de la tarea terminada. Se muestra el comentario que escribió el profesor sobre dicha entrega, en caso de haberlo escrito. La celda donde se muestra la nota, va acorde con el cuadrado que se muestra en la ventana “Lista terminadas”. Si el alumno ha aprobado la tarea, la celda es verde. Si ha suspendido es rojo y si aún no se ha calificado es de color azul claro. Al final de la lista

se encuentran dos botones, “Asignatura” y “Profesor”. Si el alumno pulsa el primer botón, se dirige a la página “Ver asignatura” que muestra la información de la asignatura a la que pertenece dicha tarea. Si el alumno pulsa el segundo botón, se dirige a la página “Ver profesor” que muestra la información del profesor que imparte la asignatura a la que pertenece dicha entrega.

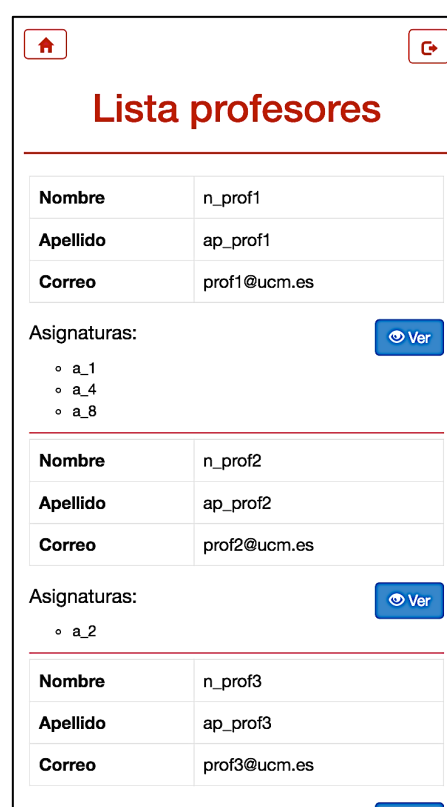
6.3.6 Ver lista de asignaturas y ver lista de profesores

Desde la opción del menú, mostrado en la Figura 36, de la página principal, se accede a la página “Lista de asignaturas” cuando se pulsa la opción “Asignaturas”, y a “Lista de profesores” cuando se pulsa la opción “Profesores”. Se pueden observar en la Figura 38 y 39, respectivamente. En la cabecera se encuentran el botón “Home” y “Salir”, anteriormente mencionados.



Lista asignaturas	
a_1	
Curso	1
Grupo	A
Media	50.0000
Ver	
a_2	
Curso	4
Grupo	A
Media	72.0000
Ver	
a_4	
Curso	2
Grupo	A
Media	50.0000
Ver	

Figura 38: Lista asignaturas - móvil



Lista profesores	
n_prof1	
Nombre	n_prof1
Apellido	ap_prof1
Correo	prof1@ucm.es
Asignaturas:	
<ul style="list-style-type: none"> ◦ a_1 ◦ a_4 ◦ a_8 	
Ver	
n_prof2	
Nombre	n_prof2
Apellido	ap_prof2
Correo	prof2@ucm.es
Asignaturas:	
<ul style="list-style-type: none"> ◦ a_2 	
Ver	
n_prof3	
Nombre	n_prof3
Apellido	ap_prof3
Correo	prof3@ucm.es
Asignaturas:	
Ver	

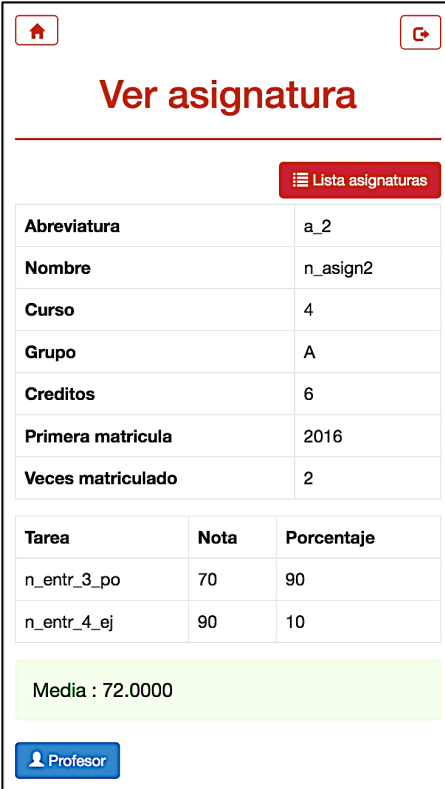
Figura 39: Lista profesores - móvil

En la página “Lista de asignaturas” se muestra una lista dividida en grupos. Cada grupo se compone de un nombre, que es la abreviatura de la asignatura, una tabla donde se muestra la información más relevante de la asignatura, incluyendo la nota media del alumno en dicha asignatura y un botón “Ver” que dirige al alumno a una nueva página donde se muestra la información de la asignatura.

En la página “Lista de profesores” se muestra una lista dividida en grupos, ordenados por orden alfabético. Cada grupo se compone de una tabla con la información de un profesor que imparte clase al alumno. Incluye una lista con las asignaturas que imparte clase a dicho alumno y un botón “Ver” que dirige al alumno a una nueva página donde se muestra la información del profesor.

6.3.7 Ver información de una asignatura y ver información de un profesor

Desde la ventana “*Lista asignaturas*” se accede a la página “*Ver asignatura*”, mostrada en la figura 40, cuando el alumno pulsa el botón de “*Ver*” perteneciente a un grupo. También se puede acceder desde “*Ver tarea pendiente/terminada*”, cuando el alumno pulsa el botón “*Asignatura*”. En la cabecera se encuentran el botón “*Home*” y “*Salir*”, anteriormente mencionados.



Abreviatura	a_2	
Nombre	n_asign2	
Curso	4	
Grupo	A	
Creditos	6	
Primera matricula	2016	
Veces matriculado	2	

Tarea	Nota	Porcentaje
n_entr_3_po	70	90
n_entr_4_ej	90	10

Media : 72.0000

Profesor

Figura 40: Ver asignatura - móvil

En la página se muestra una tabla con la información de la asignatura. En la esquina superior derecha de la tabla se encuentra el botón “Lista asignaturas” que dirige al alumno a la lista de asignaturas. Al final de la tabla se muestra una tabla con las calificaciones del alumno en dicha asignatura y el porcentaje equivalente de cada una de ellas.

Debajo de la tabla, se muestra la media de la asignatura. Si la media es apta, se muestra sobre un rectángulo verde. Si no es apta, se muestra sobre un rectángulo rojo. Si aún no se ha calificado dicha asignatura, no se muestra la tabla de calificaciones y se muestra un rectángulo de color azul claro indicando que aún no se ha calificado. Dichos colores son los mismos que se muestran en el cuadrado de “*Lista terminadas*” y en la celda correspondiente a la nota de la página “*Ver tarea terminada*”, anteriormente mencionados.

Al final de la página se encuentra el botón “*Profesor*”. Dicho botón dirige al alumno a la página “*Ver profesor*” que muestra la información del profesor que imparte dicha asignatura

Desde la ventana “*Lista profesores*” se accede a la página “*Ver profesor*” cuando el alumno pulsa el botón de “*Ver*” perteneciente a un grupo. Se puede acceder desde “*Ver tarea pendiente/terminada*” o desde “*Ver asignatura*”, cuando el alumno pulsa el botón “*Profesor*”. En la cabecera se encuentran el botón “*Home*” y “*Salir*”, anteriormente mencionados.

En la página se muestra una tabla con la información del profesor. En la esquina superior izquierda de la tabla se muestra el botón “*Lista profesores*”, que dirige al alumno a una nueva página donde se muestran la lista de asignaturas.

6.3.8 Editar contraseña

Desde la opción del menú de la página principal, se accede a la página “*Editar contraseña*” cuando se pulsa la opción “*Editar contraseña*” mencionado en el apartado 6.3.2. En la cabecera se encuentran el botón “*Home*” y “*Salir*”, anteriormente mencionados.

La página posee la misma funcionalidad y representación que la que se encuentra en el apartado 6.1.3.

6.3.9 Datos personales

Desde la opción del menú de la página principal, se accede a la página “*Datos personales*” cuando se pulsa la opción “*Información personal*” mencionado en el apartado 6.3.2. En la cabecera se encuentran el botón “*Home*” y “*Salir*”, anteriormente mencionados.

En la página se muestra una tabla con la información del alumno que está utilizando la aplicación. Dicha información incluye la facultad donde estudia y el tipo de estudio que realiza. En la esquina superior derecha de la tabla se encuentra el botón “*Editar contraseña*” que dirige al alumno a la página para editar la contraseña de la aplicación.

7. CONCLUSIONES Y TRABAJO FUTURO

En esta sección se plasman las conclusiones del trabajo realizado. Se discuten las principales decisiones sobre la arquitectura de la aplicación y de su funcionalidad. También se discute sobre el trabajo futuro de la aplicación desarrollada.

7.1 Conclusiones

Como producto final se ha creado una aplicación para dispositivos móviles que los estudiantes de la Universidad Complutense de Madrid pueden utilizar para organizar las tareas que tienen durante el período académico. La aplicación posee un desarrollo multiplataforma, por lo que es compatible con todos los sistemas operativos. En la aplicación, los estudiantes pueden ver el número de tareas pendientes y así poder centrarse en las que se encuentren más cerca de la fecha de entrega. Además pueden ver la calificación de las tareas terminadas y ver la nota media que tienen en cada asignatura en la que están matriculados.

También se han creado dos páginas web, para el administrador y el profesorado. El administrador puede gestionar las tablas de la base de datos, tales como los alumnos y las asignaturas. El profesor gestiona las tareas y su respectivas calificaciones, además ve la información relacionada con las asignaturas que imparte y los alumnos que asisten a dichas asignaturas, incluyendo las calificaciones y medias de las tareas que se presentan durante el curso.

Las aplicaciones tienen un diseño claro y limpio que resulta amigable para el usuario, cumpliendo las 10 heurísticas de Nielsen mencionadas en el apartado 5 de la memoria. Las páginas web poseen el mismo diseño, formado por tablas para facilitar la selección de los elementos. La aplicación móvil posee un diseño similar, pero orientado a un formato más cómodo para la plataforma móvil.

La arquitectura que se ha utilizado es una arquitectura cliente-servidor. Se ha desarrollado un protocolo REST, es decir, el servidor y el cliente son independientes entre sí, y por tanto se separa la interfaz de usuario y el almacenamiento de datos. El uso de dicho protocolo a condicionado los lenguajes a utilizar en el desarrollo del proyecto. PHP para el desarrollo del servidor. JavaScript, AJAX y JSON para el envío y recepción de peticiones entre el cliente y el servidor. HTML5, CSS3 y Bootstrap para el diseño de las interfaces. Además, para conseguir el desarrollo multiplataforma, se ha utilizado Apache Cordova.

La aplicación es segura mediante el uso de JWT, utilizado en cada envío y recepción de peticiones, y mediante el uso de *hashing* para garantizar la seguridad de las contraseñas de los usuarios.

Para desarrollar el proyecto, se ha creado un modelo de datos en función a las necesidades de los estudiantes en función de los requisitos estudiados, así como las operaciones a realizar por cada uno de los usuarios.

7.2 Trabajo futuro

Como trabajo futuro, se plantea mejorar ciertos aspectos de la aplicación e introducir nuevas funcionalidades. Se detallan a continuación:

1. *Integración con Moodle.* Moodle posee una serie de llamadas a su API, las cuales sería interesante estudiar y evaluar cuáles se relacionan con la funcionalidad de la aplicación.
2. *Conexión con servidor de Moodle.* La obligación de usar una aplicación para el profesorado que solo concierne a la aplicación puede resultar una tarea tediosa. Con la conexión del servidor de Moodle, solo sería necesario que el profesor introdujera los datos correspondientes a una tarea en Moodle. El sistema se encargaría de los datos para la aplicación.
3. *Desarrollo de la aplicación en plataformas nativas.* Como se menciona en el apartado 2.4 de la memoria, una aplicación con desarrollo nativo tendrá acceso a características más avanzadas del terminal.
4. *Realización de una inteligencia artificial para el control de las notificaciones.* En caso del desarrollo nativo de la aplicación, se podría realizar un estudio sobre el distinto grado de dificultad de las tareas y que se muestren las notificaciones en función a dicho grado.
5. *Ampliación de funcionalidades.* Como por ejemplo, la incorporación de notificaciones push.

CONCLUSIONS AND FUTURE WORK

In this section the conclusions of the work carried out are reflected. The main decisions about the architecture of the application and its functionality are discussed. The future work of the developed application is also discussed.

CONCLUSIONS

As final product an application for mobile devices has been created that the students of the Complutense University of Madrid can use to organize the tasks that they have during the academic period. The application has a multiplatform development, so it is compatible with all operating systems. In the application, students can see the number of pending tasks and thus be able to focus on those that are closer to the delivery date. They can also see the grade of the completed tasks and see the average grade they have in each subject in which they are enrolled.

Two web pages have also been created, for the administrator and the teachers. The administrator can manage the tables in the database, such as students and subjects. The teacher manages the tasks and their respective qualifications, also sees the information related to the subjects taught and the students who attend these subjects, including grades and averages of the tasks that are presented during the course.

The applications have a clear and clean design that is user-friendly, complying with the 10 Nielsen heuristics mentioned in section 5 of the report. The web pages have the same design, formed by tables to facilitate the selection of the elements. The mobile application has a similar design, but oriented to a more comfortable format for the mobile platform.

The architecture that has been used is a client-server architecture. A REST protocol has been developed, that is, the server and the client are independent of each other, and therefore the user interface and data storage are separated. The use of said protocol conditioned the languages to be used in the development of the project. PHP for server development. JavaScript, AJAX and JSON for sending and receiving requests between the client and the server. HTML5, CSS3 and Bootstrap for the design of the interfaces. In addition, to achieve multiplatform development, Apache Cordova has been used.

The application is secure through the use of JWT, used in each sending and receiving of requests, and through the use of hashing to guarantee the security of users' passwords.

To develop the project, a data model has been created according to the needs of the students according to the requirements studied, as well as the operations to be carried out by each of the users.

FUTURE WORK

As future work, it is proposed to improve certain aspects of the application and introduce new functionalities. Are detailed below:

1. *Integration with Moodle.* Moodle has a series of calls to its API, which would be interesting to study and evaluate which are related to the functionality of the application.
2. *Connection with Moodle server.* The obligation to use an application for teachers that only concerns the application can be a tedious task. With the connection of the Moodle server, it would only be necessary for the teacher to enter the data corresponding to a task in Moodle. The system would take care of the data for the application.
3. *Development of the application on native platforms.* As mentioned in section 2.4 of the report, an application with native development will have access to more advanced features of the terminal.
4. *Realization of an artificial intelligence for the control of notifications.* In the case of the native development of the application, a study could be made on the different degree of difficulty of the tasks and that notifications will be shown according to that degree.
5. *Extension of functionalities.* As for example, the incorporation of push notifications.

8. Bibliografía

Código desarrollado: <https://github.com/adrivent91/tfgOrganizaMisPracticas>

1. ACERCA del Campus Virtual UCM (CV-UCM). Disponivel em: <<https://www.ucm.es/campusvirtual/acerca-de>>.
2. APLICACIONES móviles UCM. Disponivel em: <<http://www.ucm.es/apps>>.
3. MÓVILES., C. D. T. P. E. D. D. A. accensit. Disponivel em: <<https://www.accensit.com/desarrollo-de-software/desarrollo-aplicaciones-app/>>.
4. API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos. bbvaopen4u. Disponivel em: <<https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>>.
5. WHAT is HTTP? w3schools. Disponivel em: <https://www.w3schools.com/tags/ref_httpmethods.asp>.
6. PAVÓN MAESTRAS, J. Ejecución del código en el servidor. Parte I: primeros pasos. Facultad de Informática Universidad Complutense de Madrid. [S.l.]. Apuntes obtenidos del campus virtual de la asignatura "Aplicaciones Web".
7. PAVÓN MAESTRAS, J. Estructura de las Aplicaciones Orientadas a Objetos El patrón Modelo-Vista-Controlador (MVC). Universidad Complutense Madrid. [S.l.]. Apuntes obtenidos del campus virtual de la asignatura "Ingeniería del Software".
8. HASHING. searchsqlserver. Disponivel em: <<https://searchsqlserver.techtarget.com/definition/hashing>>.
9. WHAT is JSON Web Token? jwt.io. Disponivel em: <<https://jwt.io/introduction/>>.
10. mamp & mamp pro. Disponivel em: <<https://www.mamp.info/en/>>.
11. [HTTPS://GOALKICKER.COM/](https://goalkicker.com/). MySQL Notes for Professionals. [S.l.]: [s.n.].
12. [HTTPS://GOALKICKER.COM/](https://goalkicker.com/). PHP Notes for Professionals. [S.l.]: [s.n.].
13. RESUMEN. Apache Cordova. Disponivel em: <<https://cordova.apache.org/docs/es/latest/guide/overview/>>.
14. [HTTPS://GOALKICKER.COM/](https://goalkicker.com/). HTML5 Notes for Professionals. [S.l.]: [s.n.].
15. [HTTPS://GOALKICKER.COM/](https://goalkicker.com/). CSS Notes for Professionals. [S.l.]: [s.n.].
16. JQUERY.AJAX(). jQuery. Disponivel em: <<http://api.jquery.com/jquery.ajax/>>.
17. INTRODUCTION. Bootstrap. Disponivel em: <<https://getbootstrap.com/docs/4.1/getting-started/introduction/>>.
18. Bringing MySQL to the web. phpMyAdmin. Disponivel em: <<https://www.phpmyadmin.net/>>.
19. 10 REGLAS HEURÍSTICAS DE USABILIDAD DE JAKOB NIELSEN. braintive. Disponivel em: <<http://www.braintive.com/10-reglas-heuristicas-de-usabilidad-de-jakob-nielsen/>>.
20. ALL-IN-ONE Prototyping Tool for web and mobile apps. justinmind. Disponivel em: <<https://www.justinmind.com/>>.
21. INTRODUCING JSON. json. Disponivel em: <<https://www.json.org/>>.